Investigation of a Command and Data Handling architecture for the SUNSAT-2 micro satellite

---

Jan-Albert Koekemoer

Thesis presented in partial fulfilment
of the requirements for the degree of
Masters of Science
in Electronic Engineering
at the University of Stellenbosch

Study leader: Prof. P.J. Bakkes

December 1999

Declaration

I, the undersigned, hereby declare that the work in this thesis is my own original work and has not previously, in its entirety or in part, been submitted at any university for a degree.

2/2/2000

Date

Abstract

This thesis investigates the design of a Command and Data Handling (C&DH) system for possible use on future SUNSAT satellites. The investigation begins with a description of the underlying components of a general C&DH system, namely telemetry (TLM) and telecommand (TCMD). From the subsequent evaluation of these subsystems on SUNSAT-1, several recommendations to improve their functionality on the proposed C&DH design are made. Preliminary design specifications and requirements are then set up for the new design. The main requirements are flexibility and reliability.

A logical improvement from the centralised TLM and TCMD designs used on SUNSAT-1, is identified, namely a bus architecture for the proposed C&DH system. Several bus technologies are subsequently evaluated and from this, Controller Area Network (CAN) technology is chosen as a suitable, relatively high speed serial bus. Several system architectures to implement CAN with on the C&DH system are then evaluated. The concept of CAN nodes within the C&DH system is explained, and from this, the detail of such a node is presented, designed, and implemented in the form of a prototype hardware system. The performance of the prototype is then evaluated against the set specifications and requirements. Other work related to the design of a C&DH system covered in this document, includes the evaluation of several international TLM and TCMD standards. From these, the recommendations of the Consultative Committee for Space Data Systems (CCSDS) are identified as a possible down- and uplink protocol on future satellites.

Opsomming

Hierdie tesis ondersoek die ontwerp van 'n bevel en dataverwerkingstelsel (Eng.: Command and Data Handling, oftewel C&DH) vir moontlike gebruik op toekomstige SUNSAT-satelliete. Die ondersoek word ingelei deur 'n omskrywing van die onderliggende komponente van 'n C&DH-stelsel, naamlik telemetrie (TLM) en telebevel (TCMD). Uit die daaropvolgende evaluasie van hierdie substelsels op SUNSAT-1, word verskeie aanbevelings gemaak ten opsigte van die verbetering van die funksionaliteit daarvan op die voorgestelde nuwe C&DH-ontwerp. 'n Stel voorlopige spesifikasies en ontwerp-vereistes word daarna bepaal vir die nuwe ontwerp met as hoofvereistes buigbaarheid en betroubaarheid.

'n Logiese verbetering op die gesentraliseerde TLM- en TCMD-ontwerpe op SUNSAT-1 word voorgestel, naamlik 'n busargitektuur vir die nuwe C&DH-ontwerp. Daarna word verskeie bus-tegnologieë ondersoek. Die toepasbaarheid van die relatiewe hoëspoed *Controller Area Network* (CAN) seriële bus word uitgewys. Verskeie stelsel argitekture word gevolglik geëvalueer waarmee CAN toegepas kan word op die nuwe C&DH-ontwerp. Ook word die konsep van CAN-nodusse binne die C&DH-stelsel uitgewys. Die detail van so 'n node word daarna behandel en die gevolglike ontwerp word dan geïmplementeer in die vorm van 'n hardeware prototipe. Die werkverrigting van die prototipe word geëvalueer teenoor die gestelde tegniese spesifikasies en vereistes. Verder word daar ook in hierdie dokument gekyk na verskeie internasionale TLM- en TCMD-standaarde. Die aanbevelings van die *Consultative Committee for Space Data Systems* (CCSDS) word geïdentifiseer as moontlik geskikte protokolle vir gebruik op toekomstige SUNSAT-satelliete.

in memory of

Christa Botes

Contents

Acknowledgements

In addition to the Lord for giving me the strength, courage and wisdom to complete this thesis, I would like to thank:

- My parents for their years of encouragement
- My study leader for his guidance and insight
- Rachel and Pierre for proofreading the document
- Hauwa for her effort in sourcing components
- The SUNSAT project for giving me the opportunity to undertake this research
- Marlene for continuous support

List of Tables

List of Figures

List of symbols

GHz	Gigahertz
Hz	Herz
KB	Kilobyte
KBaud	Kilo Baud
Kbps	Kilobits per second
m	meter
mA	milli-ampere
MB	Megabyte
Mbps	Megabits per second
MHz	Megahertz
ms	millisecond
mV	millivolt
ns	nanosecond
TB	Terabyte
μA	micro-ampere
μs	microsecond
V	Volt
W	Watt

Acronyms and Abbreviations

A/D	Analog-to-Digital converter
ACK	ACKnowledgement
ADCS	Attitude Determination and Control System
ALE	Address Latch Enable
AM	Amplitude Modulation
AOS	Acquisition Of Signal
AOS	Advanced Orbiting Systems
APID	Application Process Identifier
C&DH	Command and Data Handling
CADU	Channel Access Data Unit
CAN	Controller Area Network
CCSDS	Consultative Committee for Space Data System
CDROM	Compact Disk Read-Only Memory
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
CMOS	Complimentary Metal-Oxide Semiconductor
COTS	Commercial Off-The-Shelf
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DC	Direct Current
DIP	Dual In-line Package
DoD	Department of Defence (USA)
DVCR	Digital Video Cassette Recorder
DVD	Digital Video Disk
EDAC	Error Detection And Correction
EMC	Electro-Magnetic Compatibility
EMI	Electro-Magnetic Interference
ESA	European Space Agency
ESL	Electronic Systems Laboratory
FM	Frequency Modulation
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
FST	Flight System Testbed
GEO	Geosynchronous Earth Orbit
GND	Ground
GSE	Ground Support Equipment
HEO	Highly Elliptical Orbit

HW	HardWare
I/O	Input/Output
IBUS	Instrumentation BUS
IC	Integrated Circuit
ID	IDentification
IEEE	Institute of Electrical and Electronic Engineers
IRIG	Inter Range Instrumentation Group
ISO	International Standards Organisation
ISP	In-System Programmable
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
LONWorks	Local Operating NetWorks
LOS	Loss Of Signal
LSB	Least Significant Bit
MSB	Most Significant Bit
NIR	Near-InfraRed
NORAD	North American Aerospace Defence Command
NRZ	Non-Return-to-Zero
OBC	On-Board Computer
OE	Output Enable
OSCAR	Orbiting Satellite Carrying Amateur Radio
OSI	Open Systems Interconnection
PC	Personal Computer
PCB	Printed Circuit Board
PLCC	Plastic Leaded Chip Carrier
PLD	Programmable Logic Device
PM	Phase Modulation
PROFIBUS	PROcess FIeld BUS
PSEN	Program Store ENable
R&D	Research and Development
RAC	Remote Access Control
RAM	Random Access Memory
RCC	Range Commanders Council
RF	Radio Frequency
RFI	Radio Frequency Interference
SDLC	Serial Data Link Control
SLIO	Serial Linked Input Output
SLIP	Serial Line Internet Protocol
SO-35	SUNSAT-OSCAR 35
SRAM	Static Random Access Memory
SSB	SUNSAT Serial Bus
SSC	Synchronous Serial Channel

SUNSAT	Stellenbosch UNiversity SATellite
SW	SoftWare
TCMD	Telecommand
TDM	Time-Division Multiplex(ed)ing
TDRS	Tracking and Data Relay Satellite
TLM	Telemetry
UART	Universal Asynchronous Receiver & Transmitter
USART	Universal Synchronous/Asynchronous Receiver & Transmitter
USB	Universal Serial Bus
VCR	Video Cassette Recorder
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
WOD	Whole Orbit Data

Part A
Context

1. Introduction
2. Background

Chapter 1

Introduction

The *command and data handling system*[1] (C&DH) on any spacecraft performs two major functions: firstly it receives, validates, decodes and distributes commands to other spacecraft systems; secondly it gathers, processes and formats spacecraft housekeeping and mission data for downlink or use by an on-board computer [Larson and Wertz, 1992:380]. The first function is more commonly known as the telecommand (TCMD) system and the second as the telemetry (TLM) system. These systems may include additional functions, such as spacecraft time keeping, computer health monitoring or watchdog functions and security interfaces.

Telecommands are sent to a spacecraft in order to instruct it what to do. It therefore provides the means to reconfigure, reorient and reposition the satellite by remote control [Pisacane and Moore, 1994:601]. On the other hand, the TLM system has to monitor the configuration and health of the satellite: ie. report back on what it has measured and done. On any mission, the C&DH system must therefore exhibit a high level of integrity and reliability, whilst retaining a maximum of flexibility to respond to diverse in-flight problems or failures. In the event of failures on-board the spacecraft, the TLM and TCMD systems should be the last to succumb and their design should reflect this requirement [Mansi and Sweeting, 1987].

Before formally stating the aim of this thesis and describing the layout of the document, it is necessary to briefly introduce the context of the work done for this thesis. This is accomplished by looking at the Stellenbosch University Satellite (SUNSAT-1[2]) history and the environment in which it was built - the Electronic Systems Laboratory (ESL).

---

[1] The telemetry and telecommand subsystems on spacecrafts are referred to using various acronyms in the literature, including TT&C (Telemetry, Tracking and Command), TTMS (Telemetry, Telecommand and Modems System) and TC&R (Telemetry, Command and Ranging). Throughout this text though, C&DH will be used as a collective term and TLM (Telemetry) and TCMD (Telecommand) when referring to the individual subsystems.

[2] In this text *SUNSAT-1* will refer to the satellite in orbit today and the term *SUNSAT-2* will be used to refer to the next generation satellite; based on *SUNSAT-1* but with improved technology, redesigned subsystems and possibly different payloads.

1.1 SUNSAT-1 history

SUNSAT-1 is possibly the most complex debut satellite ever attempted by university students [Milne et al, 1999]. This is largely a result of its goal of setting a new standard in imagery from micro satellites (3456 pixel push broom imager of 15-m spacing in red, green and NIR bands [Mostert and Koekemoer, 1997]). On 23 February 1999 it became South Africa's (and Africa's) first satellite in orbit, launched from Vandenberg Air Force Base in California, USA. The programme has partially or fully supported over 96 graduate students, and is encouraging interest in technology through its Amateur Radio suite and associated schools programmes. The satellite, mounted on the Boeing-Delta II rocket inside the fairing, is shown in Figure 1.1. Figure 1.2 shows a diagram of SUNSAT-1 in flight configuration with the boom deployed. Inside the satellite behind the solar panels, is a stack of 11 trays comprising the different subsystems. These include a tray each for the TLM and TCMD systems. In the next chapter, several of the in-orbit performance results of SUNSAT-1 and lessons learnt from its operation are discussed.

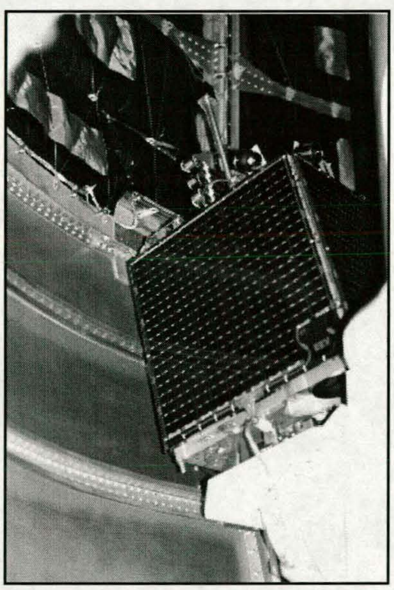

Figure 1.1 SUNSAT-1 installed on the Delta-2 rocket launcher

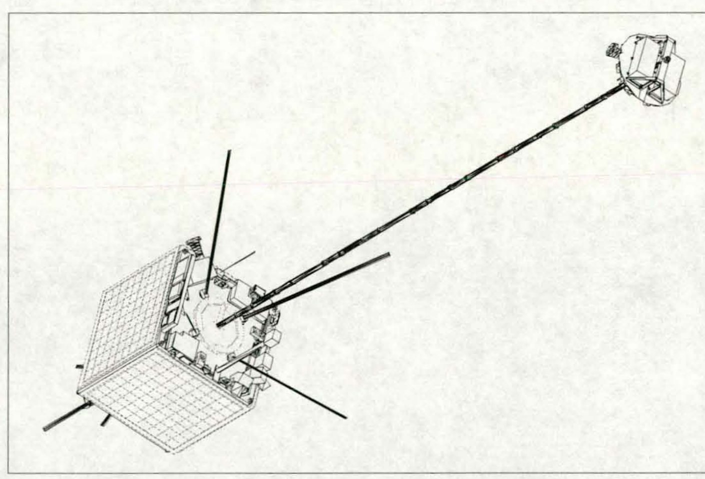

Figure 1.2 SUNSAT-1 in flight configuration

1.2 Satellite development at the ESL

The ESL was established in 1991 at the Electrical & Electronic Engineering Department of the University of Stellenbosch to serve as a facility where SUNSAT-1 could be developed and built, and also as a conducive environment for satellite research. One major difficulty experienced during the development of SUNSAT-1 stemmed from the fact that all research work had to be aimed at the completion of a flight-ready satellite. This placed several inevitable limitations on the research output.

Since the successful launch of SUNSAT-1, the ESL is in a position to broaden its horizons. On the one hand it will continue to provide a vehicle for satellite research, but it will also exploit the building of satellites as a commercially viable venture. The advantage of such an arrangement is that the research work can be aimed at improving the satellite bus and subsystems - as is the case with the work presented in this document. On the other hand, the physical development work can be based on current designs with marginal improvements. In this regard Le Roux [1998] for instance, compacted the TLM and TCMD designs on SUNSAT-1 with a resultant 50% saving in board-space. This design could be applied in the case of a satellite customer requiring increased payload volume.

All work done at the ESL can now be roughly divided into research and development (R&D), commercial satellite development, and some smaller scale industry projects. The structure of the ESL is shown in Figure 1.3 together with the breakdown of this document. The circled numbers correspond to the chapters following this one. Their contents are summarised below:

Chapter 2: Background is given on the general architecture of TLM and TCMD systems employed on low earth orbit (LEO) and other missions. The specific TLM and TCMD designs used on SUNSAT-1 are highlighted, and the shortfalls and limitations encountered during testing and operation are discussed. In addition, suggested improvements for a future C&DH design are listed.

Chapter 3: SUNSAT-1 is complex and is constructed of 11 electronic trays, interconnected by a wiring harness containing about 2000 pins [Milne, Koekemoer et. al, 1998]. A logic choice to drastically simplify the cable harness and to reduce electromagnetic interference (EMI) in the process, would be to base the new C&DH system on a bus architecture, preferably a twisted pair serial bus. This chapter therefore considers the alternative options.

Chapter 4: After choosing a suitable bus architecture, the next step is the implementation of the nodes on the bus to realise TLM acquisition and TCMD switching. Various node configurations are presented in this chapter and a suitable one is selected for the new C&DH system.

Chapter 5: A C&DH prototype implementation is presented here. Apart from conforming to the applicable specifications and relying on a suitable bus technology, two specific key improvements over the SUNSAT-1 design are highlighted in this chapter: a significantly more flexible TLM system, and a TCMD system providing 100% feedback on-board and to the ground.

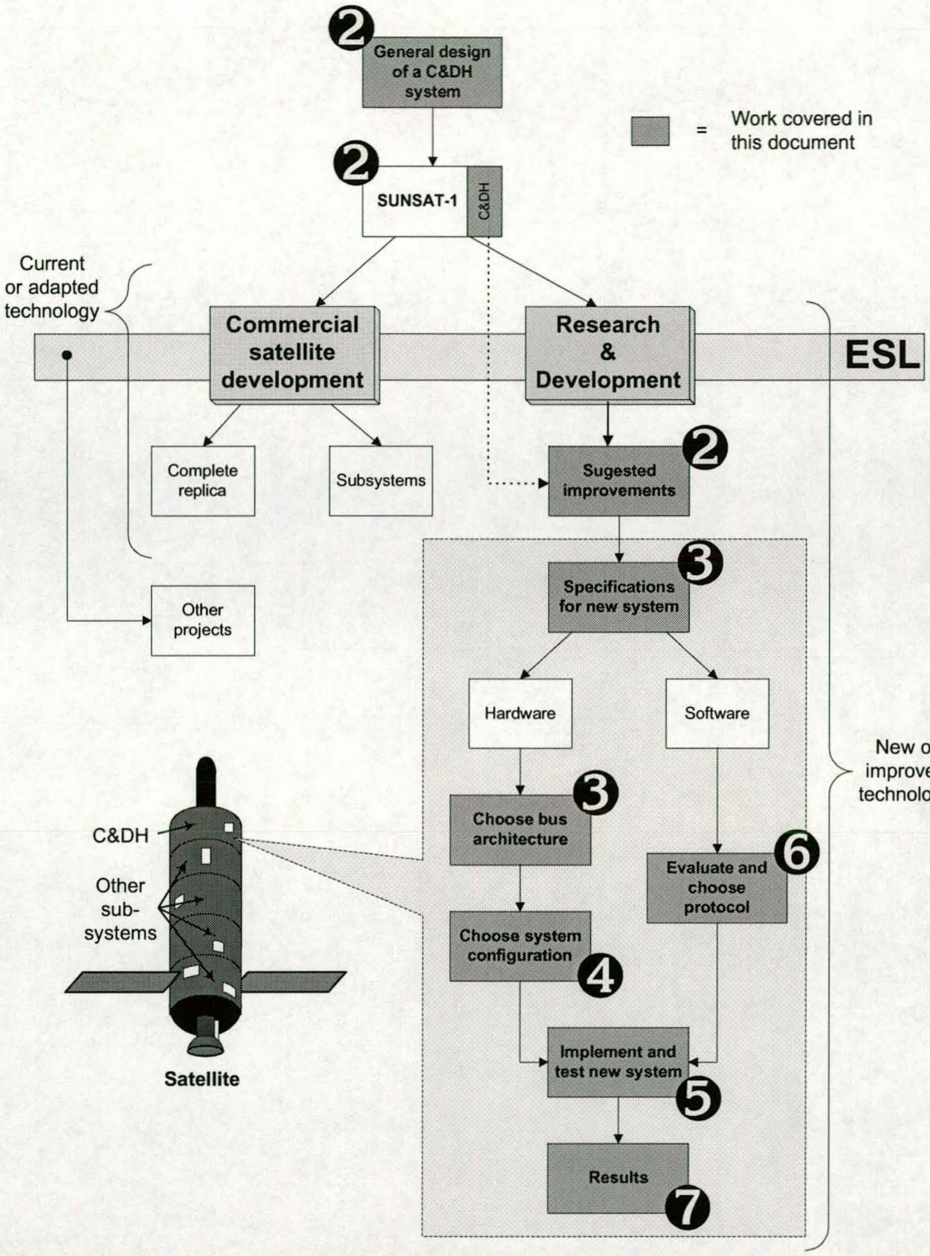

Figure 1.3 ESL structure and work covered in this document

Chapter 6: Given the chosen bus technology and system configuration above, several international protocols are evaluated in this chapter. The aim is to use one of the protocols to standardise the TLM downlink and TCMD uplink formats. Protocols evaluated include those of the Consultative Committee for Space Data Systems (CCSDS), European Space Administration (ESA) and Inter-Range Instrumentation Group (IRIG), as well as the current protocol usage in the Amateur Radio industry. The latter is to make provision for possible future orbiting satellites carrying amateur radio (OSCARs).

Chapter 7: Key aspects pertaining to the prototype board are evaluated in this chapter against specifications and guidelines set forth in previous chapters. A summary of the most important results obtained is also given.

The final chapters contain thoughts on future development, along with the concluding remarks of the document.

1.3 Aim of the thesis

This thesis investigates the design and implementation of a C&DH system for future SUNSAT missions. It is a vast improvement over the current TLM and TCMD designs employed on SUNSAT-1 in terms of efficiency, and of technology and resources used.

Although modulation/demodulation, signal conditioning, encryption/decryption, coding/decoding, and in some cases data compression, form part of the total TLM and TCMD systems used on a spacecraft; these considerations fall outside the scope of this thesis and will not be further discussed. Furthermore, the aim of the thesis is not do design a fully functional C&DH flight prototype, but rather to expand on several key aspects and lay a foundation for future research and development.

Chapter 2 Background

In this chapter, background is given regarding the design considerations for C&DH systems. The topic is introduced by means of a high-level design of a telecommand (TCMD) and telemetry (TLM) system, and the components needed to make such a systems complete and functional. The role of C&DH in different types of missions is summarised by looking at deep space, geosynchronous, and in particular, Low Earth Orbit (LEO) missions. With this background in mind; the structure, design limitations, and in-orbit performance of the C&DH system on SUNSAT-1 are evaluated.

2.1 General design considerations for C&DH systems

The ideal C&DH system is one which has previously been proven on another spacecraft and requires little or no modification for the mission under development [Larson and Wertz, 1992:380]. In practice, small steps are taken to improve the performance of the system from the viewpoint of speed, power, weight, volume or another operating parameter. In addition, strenuous testing has to be done to evaluate the capability of the system to operate despite the harsh conditions encountered during launch and in space. Where incremental design changes are not possible [1], subsystems have to be redesigned. Many of the subsystems for SUNSAT-2 fall in this category, including the attitude determination and control system (ADCS) and the new C&DH system, which is the topic of this thesis. The need for a C&DH redesign will become clear later in paragraph 2.5, when the C&DH system on SUNSAT-1 is evaluated.

The size of a C&DH system is directly proportional to spacecraft complexity [Larson and Wertz, 1992:380]. The more systems a spacecraft bus has; the more health, status monitors and configuration capability are required. Reliability concerns alone for instance, may double the size of the hardware if redundant C&DH subsystems are needed. Other factors that may influence the design of a C&DH system are the orbit and type of payload. While normally providing independent functions; the combination of command and data

[1] Considerations for a redesign - in addition to other factors - depend on the frequency of product development. For instance, when a new satellite is only developed every 10 years, technology used on the previous satellite might have become outdated and very difficult to adapt for a new design on the current satellite.

handling into a single subsystem provides an efficient means for autonomous control of spacecraft functions. An on-board computer or microprocessor can send commands and monitor telemetry over a single interface with the C&DH system, allowing control of multiple subsystems. A combined system design is also beneficial in terms of reliability, flexibility and modularity. This will become evident in Chapter 4, upon discussion of the requirements for the SUNSAT-2 C&DH system.

The following sections treat TCMD and TLM as separate logical subsystems. The components shown in each subsystem will not necessarily be realised in every spacecraft, but are discussed for the sake of completeness. It is the task of the designer to evaluate all possible hardware configurations and physical implementations; and to then select a design that most effectively suits the mission requirements.

2.2 Telecommand subsystem design

The function of a spacecraft command[2] system is to permit the spacecraft or its subsystems to be configured in response either to a radio signal sent up to the spacecraft from the ground, or to a signal received from an on-board computer. A third source of command that is not used during normal operation of a spacecraft, is a test port or umbilical cord used on the ground during testing and integration. Clearly the command system plays a vital role in the overall operation of the spacecraft [Pisacane and Moore, 1994:601].

A wide variety of commands can be sent to a satellite. Some require immediate execution, and others specify periods of delay that must elapse before their execution. Still other commands may have no strict timing requirements, simply requiring execution as soon as possible. Commands have a myriad functions: switching of subsystem power; deployment of booms, antennas, solar cell arrays and protective covers; configuration of registers and other memory components; and control of guidance and attitude control systems functions, to name but a few.

The detail of a generalised spacecraft command system and the complete TCMD segment from ground station to satellite is shown in Figure 2.1. At the ground station, new commands are assembled, encoded and sent to the command modulator, either

[2] The terms *command* and *telecommand* are used interchangeably in this text to refer to the TCMD system (see also note 1 on page 1). More specifically, *telecommand* refers to commands sent to a remote location over a specific medium, as for satellites.

automatically or by an operator. Commands may also be grouped together in the form of a macro and sent away in a batch. The modulator encodes the command message onto a subcarrier by making use of a modulation scheme such as frequency-shift-keying or phase-shift-keying. Once the subcarrier has been encoded, it is used to modulate the main radio frequency (RF) carrier by using amplitude modulation (AM), frequency modulation (FM) or phase modulation (PM) schemes. The RF link includes everything from the command transmitter on the ground to the command receiver on the spacecraft. A detailed link analysis is normally carried out at the planning stage of the mission in order to guide the design of the spacecraft antennas and the receiver sensitivity needed.

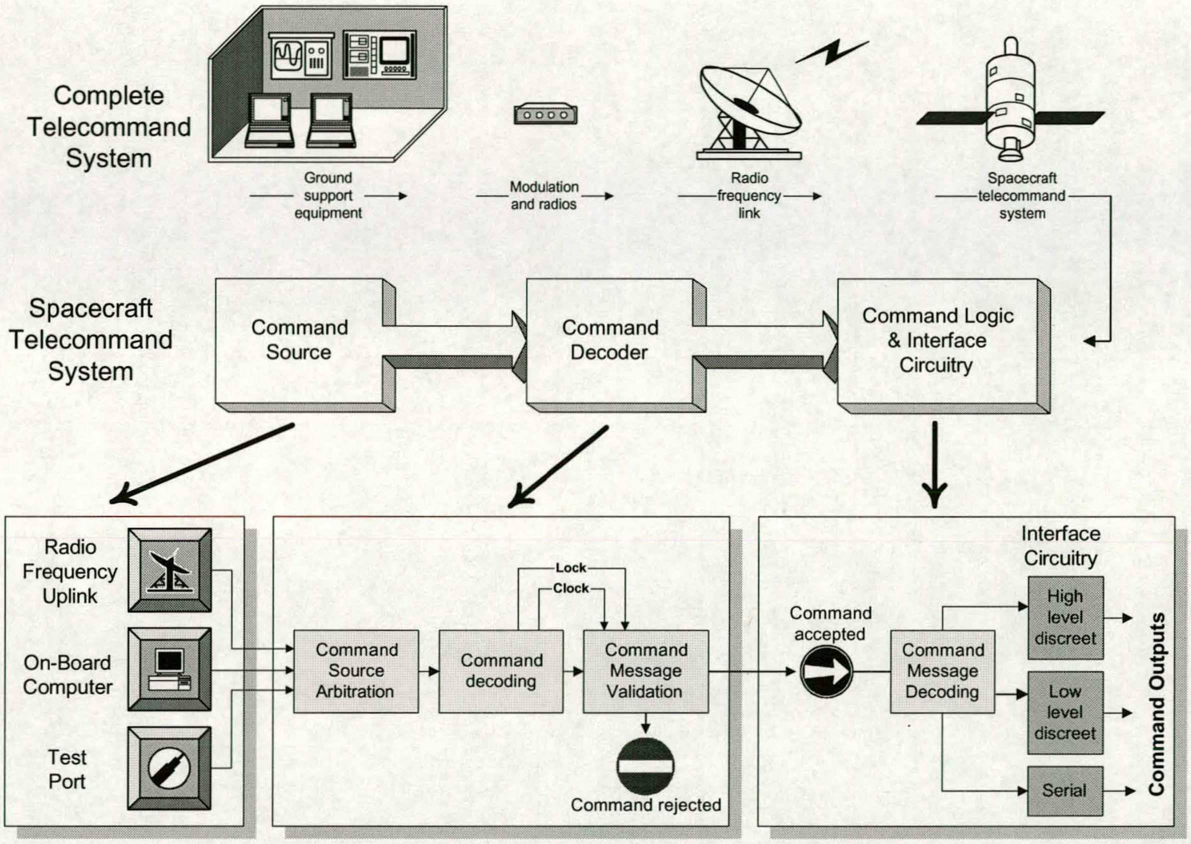

Figure 2.1 Telecommand subsystem components

Most spacecraft TCMD systems can be divided into three logical blocks: the command source, command decoder, and command logic and interface circuitry. These are discussed in more detail below.

2.2.1 Command sources

As stated above, there are normally three distinct sources of TCMDs: an uplink transponder, on-board computer (OBC) and a hardline test interface. The RF link is the most important source of TCMDs, since it acts as a 'gateway' to the spacecraft. Command receivers must therefore be extremely reliable and should be at least dual redundant. On SUNSAT-1 for instance, four different receivers can be used for command uplinking. Part of the command receiver's function is to demodulate the RF carrier and supply an amplified subcarrier to the rest of the TCMD system. The receiver must also have a sufficient degree of immunity against local interference from spacecraft transmitters and other electronics. It should have minimal power consumption, since it normally stays switched on throughout the lifetime of a spacecraft. Typical uplink carrier frequencies for TCMDs are S-band (2GHz), C-band (6GHz) or Ku-band (14GHz) although L, X, VHF and UHF bands are also used. The last two are common on amateur satellites - or OSCARs - and are also used on SUNSAT-1.

The interface between the OBC and the TCMD system can take different forms, and there are as many variations and protocols involved as there are spacecraft in orbit. The main purpose of the interface is to establish a command link to other subsystems on the spacecraft when it is not in view of a ground station. In this way, some form of autonomy is imparted on the satellite.

The hardline test interface is only important until launch of the spacecraft. It serves as an easily accessible port to the TCMD system, which can be used to issue test commands, monitor debug parameters, and verify system performance during the integration, testing, and checkout phases of the project.

2.2.2 Command decoder

As indicated in Figure 2.1, the function of the command decoder can be broken down into three different tasks. The **command source arbitration** unit makes a selection between the three different sources of TCMDs, and gives priority to uplink commands. The hardline test interface is not active during flight, and overrides the other command sources when in use.

The **command decoder** examines the subcarrier produced by the receiver/demodulator and 'undoes' the command encoding to reproduce the original command message. The message is in the form of a serial digital binary bit-stream. Most decoders will also provide two other outputs: a *lock* or *enable* signal, and a *clock* or timing signal synchronised to the serial bit-stream. The *lock* signal gives an indication to the following section that a command is about to be shifted out of the decoder on the serial data line, using the clock.

Several standards exist for command message formats (see Chapter 6). Typically, a command consists of a synchronisation code, spacecraft address bits, command message bits, and error check bits. Received commands are validated prior to execution. In some cases, critical commands are held in a register after validation while the operation code is telemetred back to the ground. Following ground validation, an execute command is transmitted to the spacecraft. If for security reasons, encryption is also added to the command message, the decryption hardware is placed between the command decoder and message validation.

Command message validation is made up of the reception of the synchronisation code, checking command message length; an exact match of the spacecraft address; an exact match of any fixed bit patterns or unused message bits; and lack of errors detected using the error check polynomial code. If the error code is sufficiently long, the command message validator should also be able to correct a fixed number of errors. It is more important for the command system to reject invalid commands than to accept and execute valid commands, since invalid commands may wreak havoc with the spacecraft and its subsystems. The probability of accepting a false command must be very low (typically in the order of 10^{-18} to 10^{-22}) [Pisacane and Moore, 1994:614]. The command message validator can also provide operational feedback via the TLM system, of the number of valid commands accepted and invalid commands rejected.

2.2.3 Interface circuitry and command outputs

The **command message decoder** interprets the meaning of each command message and reacts accordingly. The message decoder can also process delayed commands. These commands are decoded and checked, and are then loaded into a command-pending queue in memory. Either at the specified time, after a specified delay, or when a particular event occurs, the command is pulled from the queue and executed. A delayed command memory readout via the TLM system enables ground control to verify the contents of the command-pending queue.

A typical system provides two types of **command outputs**: discrete and serial. Discrete commands have a fixed amplitude and pulse duration and consist of two basic types:

- *High level discrete commands*: a +28V, 10 to 100 ms pulse used to drive a latching relay coil or fire an ordnance device;
- *Low-level discrete commands*: An open collector or 5V pulse typically interfacing with digital logic.

A special case of the low-level discrete command is the level command where a logic level is delivered instead of a logic pulse.

A serial command is a three-signal interface consisting of a shift clock; serial command data; and a data enable, used to indicate the interface is active. The **interface circuitry** has to adapt the logic signals of the command message decoder to match the requirements of the specific type of command. High-level discrete commands normally have to drive an electromagnetic coil, and therefore need sufficient drive current from the interface circuitry. In other cases, the pulse duration of low-level discrete commands are important when driving latching components.

In addition to the above components, tracking[3] may be incorporated into the C&DH system of a spacecraft, as is done on Intelsat V and Eutelsat II for example.

2.3 Telemetry subsystem design

The terms data handling and telemetry[4] are often used interchangeably. However, data handling is more than just TLM, combining TLM from multiple sources and providing it for downlink or internal spacecraft use. It can also store, or process and format the data before sending it away. In the case of a spacecraft, accurate and timely TLM is essential in order to know the detail of what is transpiring in the spacecraft and its environment. TLM data normally falls into three basic categories: housekeeping, attitude and payload. **Housekeeping data**, sometimes known as engineering data, needs to be monitored in order to keep a check on the health and operating status of the on-board equipment.

[3] A ground station sequentially modulates the TLM carrier with low-frequency sinusoidal tones. Each tone is received, demodulated, remodulated and retransmitted on the TCMD carrier by the satellite to the originating ground station. By measuring the phase difference between the original and returned tones, the station is able to determine the propagation delay, and hence the two-way travel distance. By measuring variations in this range vector over time, it is possible to establish a set of equations from which the six standard orbital elements can be derived.

Today, other less 'demanding' methods are used to obtain the orbital elements of a spacecraft. Laser reflectors (small corner-cube mirrors) on a satellite can be used to reflect a tracking ground laser beam back to its origin (as on SUNSAT-1). Apart from the widely-used North American Aerospace Defence Command (NORAD) radar tracking, a system using two Tracking and Data Relay Satellites (TDRS) can also provide tracking data coverage for 85% to 100% of most low-earth orbits.

[4] The IEEE Std 100 [1972] offers the following definition of *telemetry*. "Measurement with the aid of intermediate means that permit the measurement to be interpreted at a distance from the primary detector. The distinctive feature of telemetering is the nature of the translating means, which includes provision for converting the measurand into a representative quantity of another kind that can be transmitted conveniently for measurement at a distance. The actual distance is irrelevant."

Attitude data arises from a variety of sensors, including sun and horizon sensors, star mappers, gyroscopes, magnetometers and accelerometers. **Payload data** is variable and each case merits individual consideration. Scientific and applications payloads generally require only a few channels of data, but their rates may be very high and in some cases can justify the use of a separate, dedicated high-rate TLM system. All TLM data is either analog or bi-level discrete in origin. Alternatively, it can consist of a serial bitstream using a three-line interface.

Figure 2.2 shows a block diagram with most of the general components found in a standard TLM system. Earlier research [Koekemoer, 1997] shows that since the start of space exploration in the late 1950s, most spacecraft TLM systems were based on a time-division multiplexing (TDM) scheme. In other words, all TLM channels are sampled sequentially and in fixed order. During the past decade, there has been an effort to standardise TCMD, and in particular TLM systems, on spacecraft. From the various standards - the most important ones formulated by the CCSDS, ESA (adapted from the CCSDS standards) and IRIG - a packetised TLM system is proposed to replace the older TDM systems. The TDM telemetry system depicted in Figure 2.2 is used here to illustrate what logical components and functions are necessary to realise a standard TLM system. A packetised TLM system can be viewed as a special case of a TDM telemetry system if one regards the resulting packets as random samples of the sensors instead of sequential samples.

In Figure 2.2, as in Figure 2.1, a complete TLM system from spacecraft to ground is shown. The lower half of the figure illustrates the detail of the spacecraft TLM system. Processed TLM data is transmitted from the spacecraft to the ground in much the same way that command message data is transmitted from the ground to the spacecraft. For TLM transmission, however, the transmitter power and antenna size are much more restricted than for command transmissions. At most ground stations, the TLM-receiving antenna may be quite large. This is usually to compensate for reduced transmitter power at the spacecraft. Detailed RF link analyses must be performed in order to guarantee that the TLM system will have a sufficiently high signal-to-noise ratio, and a sufficiently low bit-error rate.

2.3.1 Telemetry acquisition

The process of TLM acquisition starts at the sensors - those devices that change their states in response to an external event or stimulus. A transducer is a particular type of

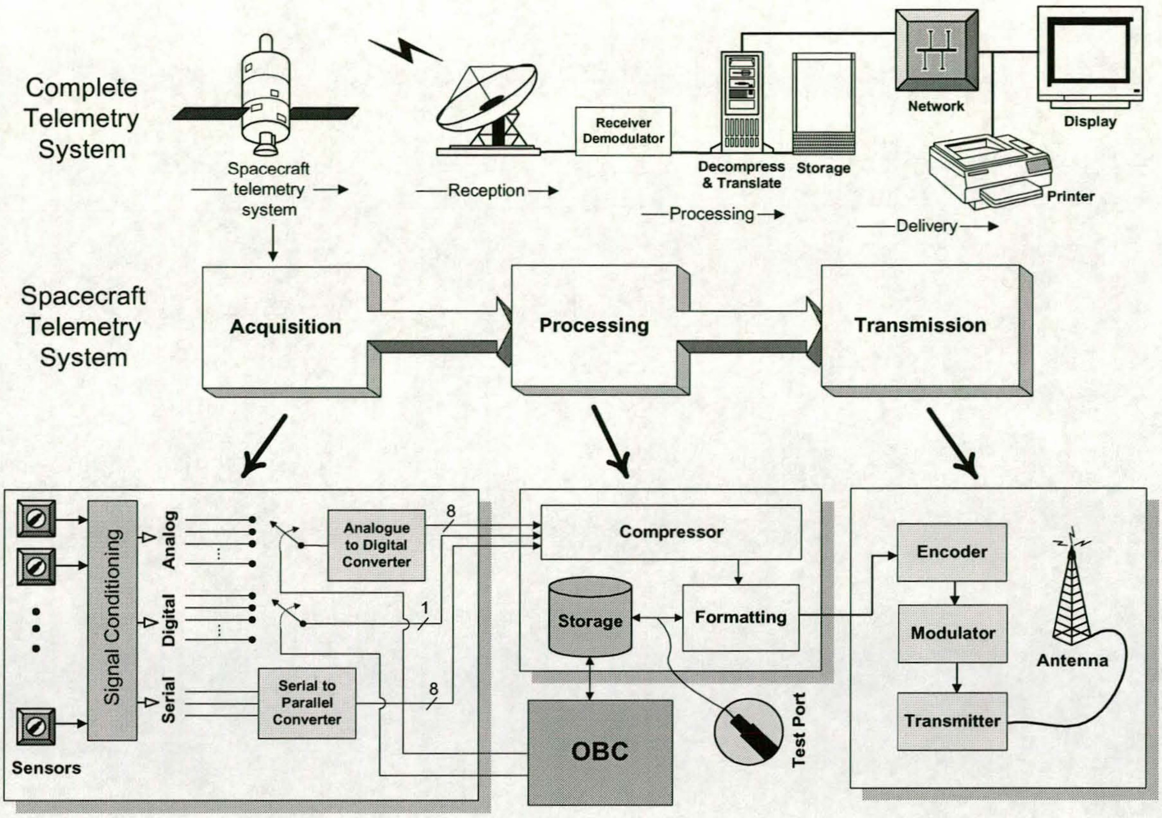

Figure 2.2 Telemetry subsystem components

sensor that converts one form of energy into another. Certain phenomena are more difficult to measure than others and require more complex sensors. Magnetometers, altimeters and radiometers fall in this class and are frequently designed as entire spacecraft subsystems. The parameters measured most frequently in a spacecraft are voltage, current and temperature. This requires only simple electronics.

The process of converting raw sensor outputs to voltages in a standard range is called **signal conditioning**. The frequency response, impedance, ground reference and common mode range of the signal conditioning electronics should all meet the requirements of the signal under measurement. Care should be taken to ensure that the signal is not deformed or altered before it is actually measured.

In the TDM TLM system at hand, signals that have been conditioned are selected one at a time by a multiplexer. This examines only one particular voltage from a field of many. One commutation[5] cycle forms a frame of TLM data and each piece of data is sampled at least once per frame. The frequency of the commutator must be the same or higher

5 Usually a multiplexer steps through the conditioned voltages in a fixed sequence of rotation. Such a multiplexer is then called a commutator.

than the Nyquist frequency of the highest frequency signal to be sampled. In practice, the sampling frequency is typically five or 10 times higher than the highest frequency of interest. The TLM system can also be set up under command from the OBC for instance, in such a way that only certain sensors are sampled or so that several sensors are sampled more often than others. This is known as a **dwell function.**

An **Analog-to-digital converter** (A/D) is used to change each conditioned analog signal to a digital equivalent before it is processed. The input voltage range to the A/D is typically 0 to 5V (unipolar) or -2.5 to 2.5V (bipolar). Many TLM systems only use eight-bit A/Ds resulting in a conversion sensitivity of 1 count per 20mV. The conversion speed and power consumption of the A/D device can play a significant role when the sampling rate is very high.

2.3.2 Telemetry processing

Once the TLM data has been converted into digital form, it can be processed before transmission to the ground. This reduces the number of data bits that must be transmitted and thus reduces the required bandwidth of the TLM transmitter. The process is known as **data compression**. On-board TLM data processing also helps to augment spacecraft autonomy. The processor can analyse the TLM data and inform the command system automatically if corrective action is needed.

Following collection and processing of the data, the TLM system adds some or all of the following information:

- A synchronisation word - indicates which bit is the first in the frame;
- A frame count - shows which frame is currently in transmission;
- The spacecraft ID - indicates which spacecraft is the source of the TLM data;
- Error detection/correction bits - enables errors in each transmitted block of data to be detected and corrected at the ground station;
- A frame format ID - indicates which of several formats is in use; and/or
- A time tag - states when the TLM frame or packet was assembled.

The process of adding this information to the TLM data is called **formatting**. For satellites in low-earth orbit and inter-planetary missions, continuous data coverage is not always possible. In such cases, TLM data must be **stored** on a mass-storage device for later retrieval when the satellite is in view of the ground station. Such data is referred to as whole-orbit data (WOD).

As is the case with the command system, a **test port** is provided for easy accessibility to the TLM data during integration and testing. This allows higher data rates, but RF links are still used for overall performance testing.

2.3.3 Telemetry transmission

TLM data is sent to the ground station over an RF link similar to that used by the command system. The most common type of modulation used is phase shift keyed-pulse code modulation (PSK-PCM) [Pisacane and Moore, 1994:624]. The use of PCM facilitates data encryption and the implementation of error detection and correction (EDAC) codes. At higher bit rates, phase modulation is also used. Transmitting energy at a rate of 12W using a carrier RF of 2GHz may draw energy from the power system at a rate of 60W, for instance where class-C amplifiers are used in the transmitter. That is, for each watt transmitted, 4W are wasted. It is thus desirable to keep spacecraft transmitter power low and to ensure that unnecessary information is not transmitted.

2.4 TCMD & TLM adaptations for different orbits

Low Earth Orbit (LEO), Geostationary Earth Orbit (GEO), Highly Elliptical Orbit (HEO) and interplanetary or non-geocentric missions cover most of the orbits in use today. Some of the requirements they impose on TCMD and TLM systems in general are highlighted in Table 2.1.

It is evident from the table that all of the attributes listed could be applied to LEO satellite C&DH design due to the short times for which these satellites are visible to the ground station. Delayed commands and data storage are necessary for remote parts of orbit, as is spacecraft autonomy. Fast downlinks for stored payload data and rapid commands during visible passes are essential for effective operation of a LEO satellite.

2.5 SUNSAT-1

Based on the material presented in this section about the C&DH system on SUNSAT-1, a corresponding but improved prototype C&DH system will be designed in the following chapters. It is therefore important to gain some understanding of the design and performance of the TLM and TCMD subsystems on SUNSAT-1. Some of the in-orbit results are presented, as well as the lessons learnt during the integration, launch, and operation of SUNSAT-1 since February 1999.

Table 2.1 TLM & TCMD requirements for different orbits

Attribute	LEO (UoSAT) Note 1	GEO (Intelsat) Note 2	HEO (ISO) Note 3	Interplanetary (Mars surveyor) Note 4
Delayed commands	✓✓		✓	✓
Rapid commands	✓			
Data compression	✓			✓
Data storage Note 5	✓✓			✓✓
Coding Note 6	✓	✓	✓	✓✓
Bit rates Note 7	1mbps, 9.6kbps	Telephony	32kbps	21.33kbps
Error correction	✓	✓	✓	✓✓
Frequencies Note 7	L,S-band UHF,VHF	C-band Ku-band		X-band Ka-band
Spacecraft autonomy	✓✓			✓
Radiation-hardened components	✓✓ Note 8	✓	✓✓	✓ Note 9

A ✓✓ indicates a high probability and a ✓a lower probability of use.

Note 1: UoSat-12 was launched in April 1999, weighs 325kg and was built by SSTL in the UK - their first mini-satellite. It contains various earth imaging and communications experiments.

Note 2: The four Intelsat-8 satellites are operational since May 1999 and provide global telephony, video, and other business services.

Note 3: The Infrared Space Observatory (ISO) was launched in Nov. 1996 into a 1000km x 70500km orbit, was operational for 2 years and contained an infrared camera and various other astronomical instruments.

Note 4: Mars Global Surveyor was launched in Nov. 1996 and weighs just over a ton. It studies various geological and morphological parameters of the Martian surface and atmosphere.

Note 5: The type of storage device is dependent on the mission and payload requirements, but most LEO satellites today use solid state memories. Tape recorders are being used on interplanetary missions.

Note 6: Coding depends on the link budget of the mission, but it is considered mandatory on deep space missions where the extra gain results in a saving of transmitted power - a scarce resource on such missions.

Note 7: These parameters are highly dependent on the type of mission and payloads included. The values listed correspond to the particular examples given.

Note 8: Although LEO orbits justify the use of radiation hardened components, budgetary constraints normally limit the use of such components on micro satellites. UoSat-2, for example, is still operational since launch in 1984 and uses commercial components.

Note 9: The total accumulated radiation dose per year on an interplanetary mission can be more than two orders of magnitude less compared to a spacecraft in LEO [Piscane and Moore, 1994:698]. However, the cost of the mission justifies the use of rad-hard components on these missions.

As previously stated, SUNSAT-1 consists of a stack of trays containing the subsystem electronics. The centralised TLM and TCMD systems, with a fair amount of built-in redundancy, each occupy one full tray in the satellite. Their designs are discussed below.

2.5.1 Telemetry subsystem design

When the satellite project was started at the University of Stellenbosch in 1992, an initial technical design specification (tech-spec) was formulated. Reliability was determined as the main requirement for the TLM system of the satellite [De Swardt, 1994:18]. Secondary requirements include [De Swardt, 1992]:

- Implementation of both a hardware and software based TLM system;
- Flexibility to adapt to different situations;
- Collection of both analog and status telemetry data;
- Implementation of a WOD facility;
- Implementation of a dwell facility;
- Provision for 248 analog and 184 status channels; and
- Use of amateur packet radio formats and frequencies to transmit TLM data.

It was also decided to build seven 1200-Baud and three 9600-Baud modems on the same tray as a logical extension of the TLM system. The resultant dual-redundant, time-division multiplexed TLM hardware implementation is shown in Figure 2.3. The TLM system is divided into three sections, labeled systems A, B and C respectively. For redundancy purposes System C is an exact replica of system B. System A functions identically to system B and C, but is implemented in software using a micro controller (80C31) as an extra measure of redundancy. There are eight other major subsystems on the satellite, each containing two identical 32-channel analog acquisition units (analog multiplexers) and a 24-channel status acquisition unit (digital multiplexer). Each of the resultant 256 (8 x 32) analog channels are sampled consecutively by the 8-bit A/Ds contained within system B or C at a rate of either 0.39Hz or 3.13Hz selected by TCMD. The status channels are combined with the sampled analog data by replacing the LSB of the first 24 analog samples on each subsystem with the corresponding status bit.

A sequence of 256 bytes, representing the status of 256 TLM channels, is referred to as a TLM frame. Data frames generated in this manner are serialised by a UART on each system (B & C) for transmission to the ground. Two 1200-Baud modems (one each for systems B & C) are used to modulate the bit stream. Systems B & C interface to the OBCs via an 8-bit parallel bus using two interrupt lines to indicate the arrival of a valid data byte (INT_TLM) and the end of a valid frame (INT_SEL). System A is not as rigid as systems B & C, and can implement a dwell function to sample only a set of specific TLM channels. This command is implemented by the SUNSAT Serial Bus (SSB). The sampled TLM data originating at Systems B & C is converted to a serial bit stream by the 80C31, and is then transmitted to the OBCs via the SSB. A 32KB memory connected to the 80C31 can be used to store the sampled telemetry data temporarily. The same 80C31 is also used for keeping a list of TCMD switches and their states - refer to the following section.

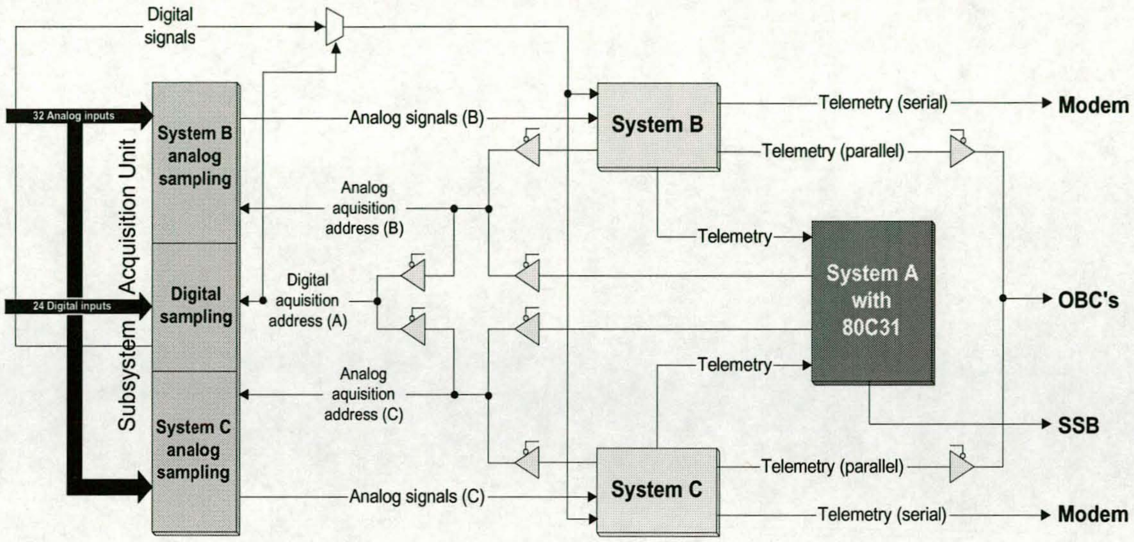

Figure 2.3 SUNSAT-1 telemetry subsystem

The board space for the SUNSAT-1 TLM system is occupied by the system A, B & C electronics; the 10 modems; and various analog and digital multiplexers in order to route the different signals between the radios, modems and OBCs.

2.5.2 Telecommand subsystem design

As is the case with the TLM subsystem, a set of initial performance requirements was set forth in the SUNSAT technical specification document for the TCMD system [Weber, 1992]. Thereafter, the first version of the board was designed and tested. The primary requirements were [Botha, 1994]:

- Sufficient command latching capability - 256 latches were initially provided;
- Small false command probability - one false command in ten years allowed;
- Fast throughput rate - no command should take longer than one second to process;
- Multiple inputs - four modems and two on-board computers;
- Reliability - the system should be operational at all times; and
- Restricted access - all commands should be received in duplicate and encapsulated in a valid 16-bit password before execution.

Figure 2.4 shows a high-level block diagram of the SUNSAT-1 flight-model TCMD system. The multiple source requirement is met with two 9600-Baud modems and two dedicated 1200-Baud modems, the parallel OBC interfaces, as well as a serial bus under command from the OBCs. The 9600-Baud modems can also be used to receive general purpose data. Two subsystems as indicated, each with its own power supply regulators, are used for redundancy. Each subsystem has 232 switches of which 128 are bit-addressable and the rest byte-addressable. These switches are combined using XOR gates to ensure that the state of a command line to a subsystem can still be changed if any one input to a specific XOR gate becomes stuck at one logic level. A further measure of redundancy is implemented by interfacing the 1200 modem on subsystem 1 to discrete hardware, and the 9600 modem to a 8031 micro processor; this process is reversed for subsystem 2.

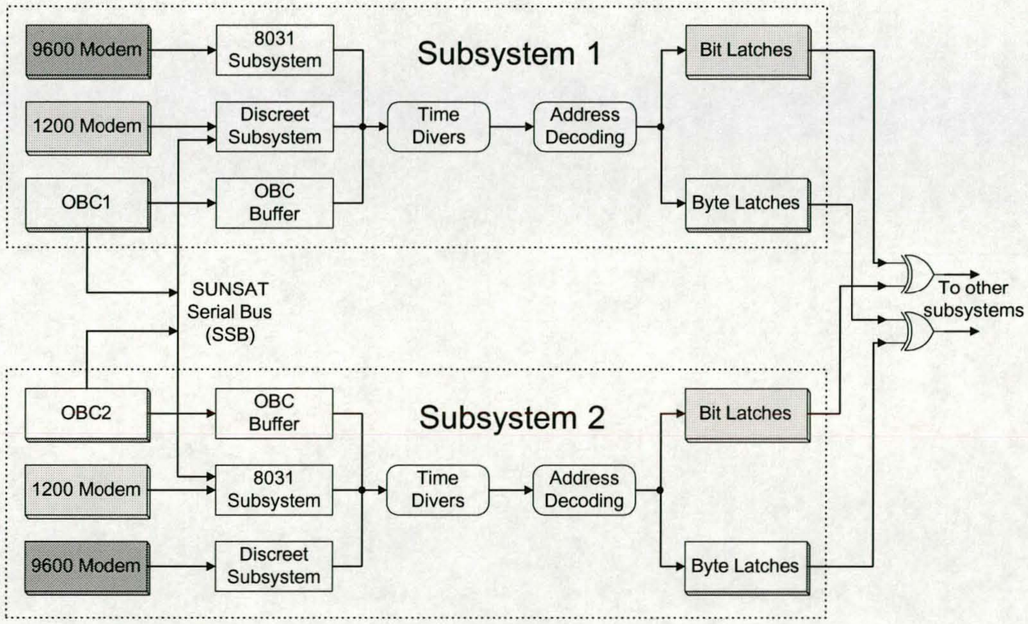

Figure 2.4 SUNSAT-1 telecommand subsystem

A timing diversity system forces each TCMD to be sent to the satellite twice before execution. The address decoding section checks for a valid 24-bit password (increased from 16 in the original specification in order to ensure higher level of security) at the beginning and end of each received command. It also validates the parity byte before interpreting the address and data bytes. Each valid command therefore consists of 72 bits. In order to change the state of a switch or group of switches, a 144-bit data stream must be transmitted to the satellite by the ground station. The OBCs, however, send only three bytes to the TCMD system: a data, address and password byte. As before, this must be done twice.

2.5.3 Lessons learnt

The SUNSAT project started from scratch in 1992 and evolved into a big research project with a working satellite in orbit in 1999. In this period, many original goals and specifications have changed. The continuity of the project has also been hampered over time by the graduation of Masters students at the end of each two- or three-year period. The design of the SUNSAT-1 C&DH system is thus by no means optimal or 'state-of-the-art'. Specific problems, shortfalls, and limitations do exist. Those most important, pertaining to the TLM and TCMD subsystems and identified during testing, integration and after launch are listed below.

Telemetry

- The main problem with this **TDM** system is its inflexibility: each subsystem on the satellite is assigned 32 analog and 24 status channels regardless of respective requirements. The fact that many unused channels are also sampled in every frame leads to the **ineffective use of bandwidth,** and thus a loss a valuable power during direct transmission to ground, and a loss of storage space during WOD collection [Koekemoer, 1997].
- Due to the centralised implementation of the system, the **cable harness** is very large and **complex**: five address lines are routed to every subsystem and the individual outputs from each are combined at the TLM board.
- The **UARTs** on the TLM system can **only** transmit TLM at **1200-Baud** in real-time downlink mode. Although the system is capable of a 9600-Baud sampling speed, this is only applicable on the satellite itself.
- When the OBC is collecting WOD from the TLM system, it is interrupted every 10ms[6] when the system is running at 1200-Baud or every 1.25ms[7] when running at 9600-Baud. Under normal on-board operation at 1200-Baud, TLM interrupts account for half the OBC interrupt load. This **interrupt burden** further increases the workload of the slower first or main OBC, called OBC1.

---

[6] At 1200-Baud, provision is made for 12 bits per symbol. Therefore every symbol (or sampled TLM value) arrives at: $\dfrac{1}{\dfrac{1200\,\text{Baud}}{12\,\text{bits}}Hz} = 0.01s$

[7] At 9600-Baud, provision is also made for 12 bits per symbol. Therefore every symbol (or sampled TLM value) now arrives at: $\dfrac{1}{\dfrac{9600\,\text{Baud}}{12\,\text{bits}}Hz} = 0.00125s$

- Although it is not always necessary to have the TLM system running, it **cannot be switched off** or commanded into a power-down or low-power mode. The system constantly draws approximately 130mA from the satellite batteries.

Telecommand

- The use of the **XOR gates** at the output of the TCMD system has one major drawback: if TCMD subsystem 1 sets one of the inputs to an XOR gate and TCMD subsystem 2 subsequently sets the other input, the output of that gate will be reset. To alleviate this problem, the two on-board computers have to be aware of each other's actions as far as addressing the TCMD system is concerned.
- The other major problem stems from the first. Originally, it was intended that any change in the TCMD switches be reflected in the TLM data. However, the majority of control lines on SUNSAT-1 do not have any effect on TLM. This **lack of feedback** makes it very hard to assess the current state of TCMD switches.

2.5.4 In-orbit performance

No system can be evaluated to it's full extent until 100% operational in it's intended environment. This applies particularly to a space environment where the effects can only be simulated up to a certain point. One can also not help but become spoilt by the proximity of a satellite in a lab environment. Only once the satellite is irrevocably and physically separated from earth up in space, does one begin to appreciate the importance of feedback mechanisms, secure communications and most importantly, reliability. The primary requirement for a debut satellite is to operate in space, and if unavoidable, to fail "gracefully"! [Milne et al, 1999]. Until the time when this document was submitted, SUNSAT-1 was still operational in space, meeting, and in some cases exceeding, its performance expectations. Orbital results specifically related to the TLM and TCMD systems are reported below.

Telemetry

The first sign that SUNSAT was working in space, 12 hours after its launch on 23 February 1999, was the TLM received at the Stellenbosch ground station. Currently, after more than nine months in orbit, the TLM transmitter is still switched on manually from the ground station during most visible passes at acquisition of signal (AOS), as a means to confirm that SUNSAT is still "alive". This underlines the importance of TLM as a primary feedback mechanism on any satellite.

To date, system B on the TLM system has been operating flawlessly. It is running exclusively at 1200-Baud, as the need for a higher sampling speed has not yet arisen. Neither System C nor the 8031 system have been tested or used. The real-time TLM Z-axis magnetometer readings in Figure 2.5 show how the angular rate normal to the Z-axis dropped by a large factor as the boom deployed. This particular TLM trace confirmed that the ESL-developed boom had deployed successfully [Milne, Koekemoer et al, 1999].

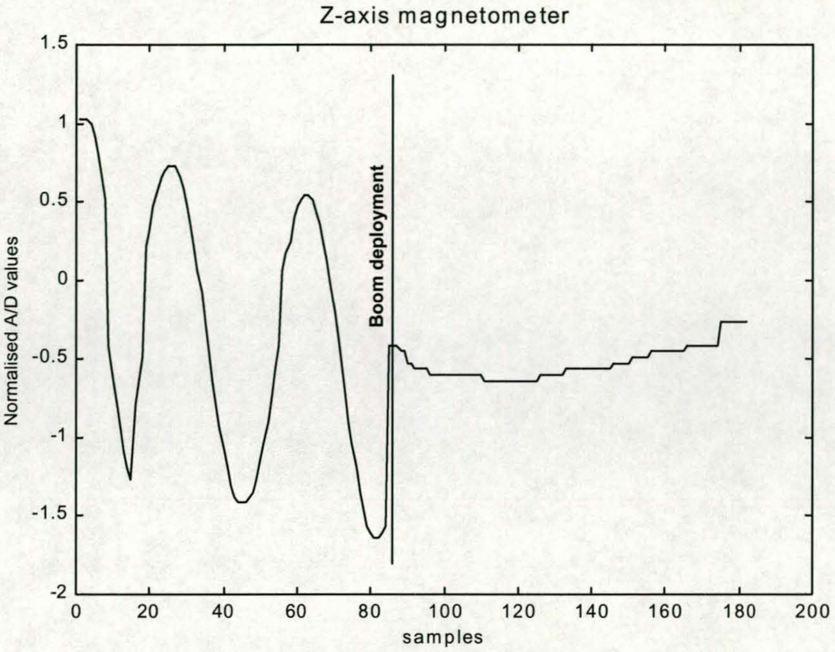

Figure 2.5 Z-axis magnetometer reading before and after boom deployment (26/3/1999)

WOD is used extensively during the mission to verify system parameters during remote parts of orbit. Figure 2.6 shows solar panel temperature WOD collected during seven continuous orbits.

Telecommand

During the first five months of operation, approximately 110 000 TCMDs[8] have been transmitted from the Stellenbosch ground station to SUNSAT-1. Without a hard feedback mechanism on the satellite, it is very difficult to determine the success and failure rates of these TCMDs. During this five-month period however, no known false or erroneous TCMDs have been reported. A more accurate test would be to program the OBC to update a history file when it detects a ground TCMD. This file can then be compared to a session log file at the ground station to determine any discrepancies.

[8] This is not an exact number, since no system was in place during the two weeks following the launch to keep a log of all ground TCMDs.

To combat the difficulty encountered with two different subsystems and the XOR gates, a status list of TCMDs is kept on the 8031 RAM on the TLM board. Both OBCs then make a copy of this list in local RAM via the SSB when it starts up. The list is updated whenever a TCMD is set from the ground or by an OBC itself. Caution must still be exercised when setting TCMDs from the ground and both OBCs are switched off.

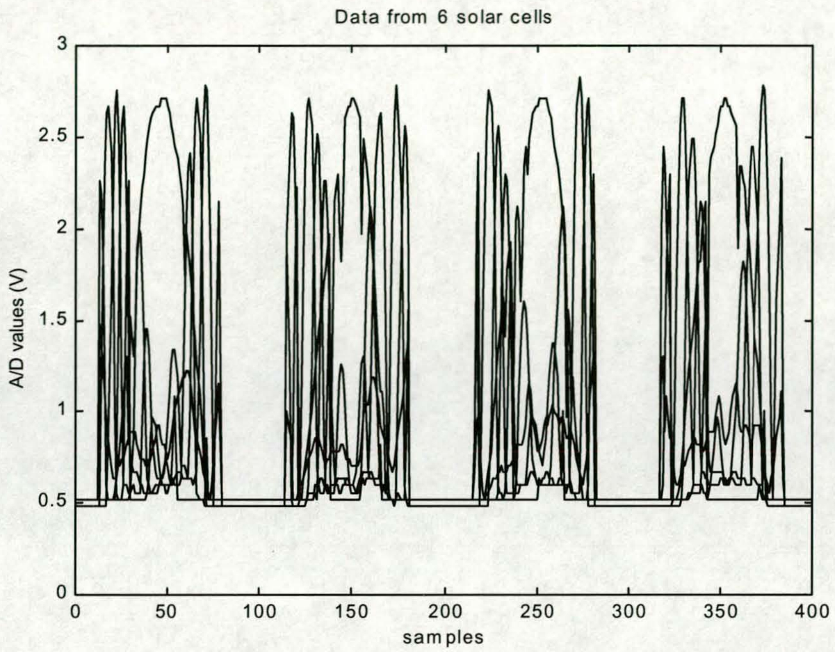

Figure 2.6 Course solar cell Whole Orbit Data over 4 orbits (9/8/1999)

2.5.5 Suggested improvements

The proposed improvements suggested below follow from the lessons learnt:

Telemetry:

- *Flexibility* - a degree of freedom should exist to add and remove TLM channels when necessary. This will optimise the use of bandwidth.
- *Simplified cable harness* - the myriad of wires connecting the different subsystems to the centralised TLM and TCMD systems must be reduced, preferably by a bus architecture with only two or three wires interconnecting all subsystems on the satellite.
- *Variable sample speed* - although this is already achievable on SUNSAT-1, depending on the link budget it should also be transmitted to ground at higher speeds than 1200-Baud, given the small window of operation per visible pass.
- *Improved OBC interface* - the C&DH system should not burden the OBCs with interrupts. It should have either a dedicated interface to a mass storage area or in the case of WOD collection, be able to buffer data before passing it on to the OBC.
- *Better power consumption* - given that the TLM system presents a constant or high load

on the power system, it should be able to switch into a low power or power down mode when not required to be fully operational. In any case, the system should not on average consume a high amount of power.

Telecommand:

- *Less board space* - currently the XOR gates on the SUNSAT-1 TCMD system occupies a lot of board space. The system design should be evaluated in order to improve this situation.
- *Improved feedback* - all telecommands on the satellite should have either a corresponding result on a TLM channel, or be reported to the ground or the OBCs via an effective feedback mechanism.
- *Better power consumption* - the TCMD system will always be switched on and should therefore consume a minimal amount of power.

2.6 Conclusions

The broad requirements for building a satellite telemetry and telecommand system and how it should be adapted for different orbits have thus been addressed. The C&DH system on SUNSAT-1 was presented as an example, albeit not a perfect one. Shortfalls and limitations identified during its integration and operation have been discussed. The suggested improvements following from this provide a backdrop for the design of a new C&DH system; building on the SUNSAT-1 experience, yet incorporating a much-needed revamp.

Part B
D e s i g n

3. Choice of bus architecture
4. Structural design of new C&DH system
5. Implementation of the test setup
6. International TLM & TCMD standards

Chapter 3

Choice of bus architecture

The fundamental requirement in all network applications that involve two or more functional nodes is the provision of a suitable communications facility. As stated in the previous chapters, the parallel TLM and TCMD bus on SUNSAT-1 presents various problems in terms of cable harness complexity and EMI. This chapter therefore explores the different serial communication media and protocols used to facilitate reliable operation of the new C&DH system.

3.1 Communication medium requirements

The type of communication medium used in any system is a function of [Halsall, 1992]:

- **The nature of the application**.
  The nature of a satellite network application is dictated by the physical separation between the satellite and the ground station as well as the harsh radiation environment encountered in space.
- **The number of functional nodes involved**.
  For future SUNSAT micro satellites, the number of C&DH nodes[1] would be of approximately the same order as the combined total of subsystems and payloads. At this stage, a reasonable node estimate would be between 10 and 20, taking into account the number of subsystems on SUNSAT-1.
- **The physical separation between nodes**.
  The dimensions of a micro satellite are approximately $30cm^3$ to $80cm^3$ (SUNSAT-1 is a $45cm^3$ satellite). The resulting physical separation between any two adjacent nodes would not be more than approximately 30cm, excluding the distance between a tipmass and the main satellite body (usually between 2m and 8m).

Taking these broad guidelines into account, the following sections evaluate several serial communication architectures. A comparison is then drawn in order to identify a suitable architecture for inclusion on the proposed C&DH system.

---

[1] The concept of nodes, the logical implementation of the new C&DH system and supporting arguments are presented in the next chapter. For the purpose of this chapter 'node' should be viewed as the data interface point between the C&DH system and any other subsystem.

3.2 Evaluation of possible bus technologies

This section discusses the various bus architectures available for implementation in the C&DH system. Only serial architectures are considered, since the parallel architecture has already been shown to be unsuitable for this application due to the complexity of the cable harness.

3.2.1 SUNSAT-1 buses

Two of the prominent serial buses in use on SUNSAT-1 are discussed here as background. Later in the chapter they are used for comparison with the other buses considered below. Figure 3.1 shows the two serial buses on SUNSAT-1 and the subsystems which they interconnect.

SUNSAT Serial Bus (SSB)

The SSB is one of two in-house serial buses and was developed for low speed (9600-Baud) communication between the OBCs and the TLM, power systems and two TCMD micro controllers (80C31s) [Le Roux, 1995]. A proprietary one-line, transistor-coupled interface plus *ground* is used to connect the standard UARTs on six different processors. The architecture is a half-duplex bus, and the access method is master-slave. This means that either of the OBCs, but not both simultaneously, is the master; and the TLM, power system, and TCMD processors are the slaves. The protocol used is Serial Line Internet Protocol (SLIP) [Romkey, 1988] and the maximum length of a frame is 256 bytes. Each frame is encapsulated by two <END> characters (decimal 192), and byte stuffing is undertaken whenever an <END> or <ESC> (decimal 219) character in the data portion of the frame is encountered.

The SSB was implemented to facilitate the transmission of dwell TLM packets from the 8031 TLM system processor to the OBCs. It also sets up and receives status information from the power system 8031 processor, and sets TCMDs from the OBCs. The SLIP protocol does not contain any error detection and correction, or compression algorithms. As a result, a simple error checking byte - the XOR of all preceding bytes - is included at the end of each frame before the <END> character.

SUNSAT Instrumentation Bus (IBUS)

The IBUS is a higher speed (19.2KBaud) serial bus and is connected to the OBCs, the tipmass and the GPS/school experiments tray [Steenkamp, 1996]. The purpose of the bus is to allow the on-board computers to send commands and receive data from the tipmass instruments and school experiments. The physical medium is a dual-redundant RS-485 bus. Half-duplex communication is used with multiple access and collision detection. An IBUS manager is implemented on the OBCs. To minimise multiple collisions, each IBUS node has a unique and fixed delay-time. It must wait for the specified period after the previous bus activity before beginning transmission. The proprietary protocol uses one header byte to indicate both the target node and the number of data bytes transmitted - up to a maximum of 31 bytes. A trailer byte is used for a modulo-256 checksum. In-between is the command/data ID and the data section. All nodes answer a received packet with an acknowledge (ACK) or a negative acknowledge (NACK). Retransmissions are implemented in the case of a NACK or time-out. During the configuration mode (after power-up) the IBUS manager transmits an èecho'command. From the replies received on both buses, either IBUS1 or IBUS2 is selected for communication.

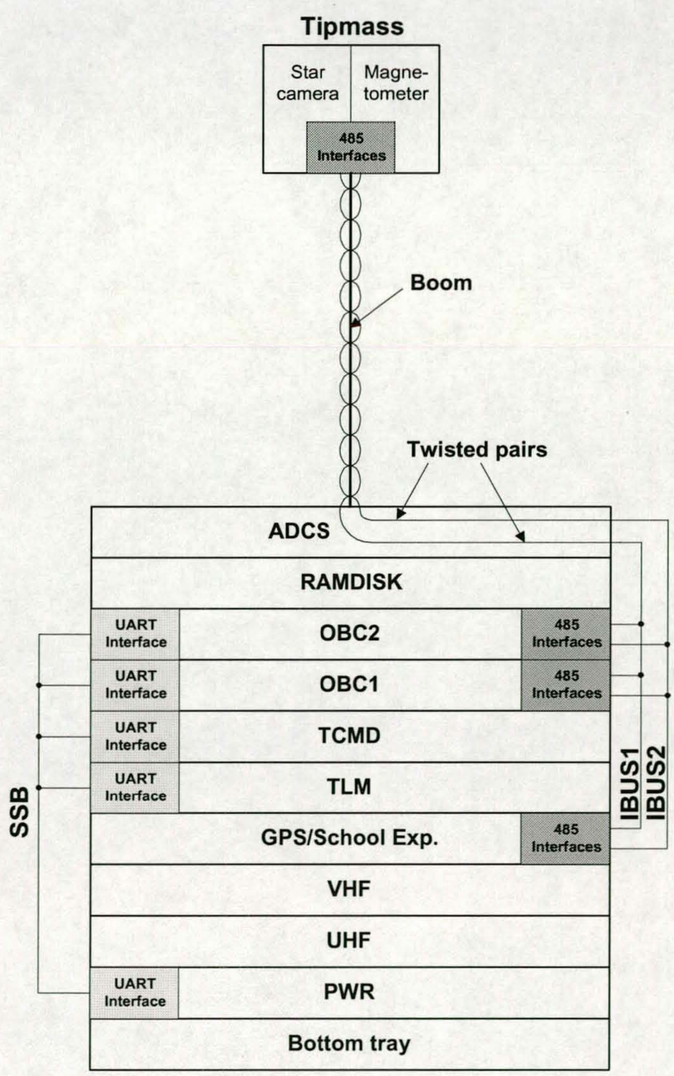

Figure 3.1 SUNSAT-1 serial buses

The IBUS primarily facilitates reliable, medium-speed communication between the OBCs and the magnetometer and star camera on the tipmass. The latter is physically separated from the rest of the satellite and is therefore prone to external electromagnetic disturbances due to the long wires running up the boom. Consequently, RS-485 drivers and shielded, twisted-pair wire are used.

3.2.2 Consumer buses[2]

The two major consumer buses discussed here are the Universal Serial Bus (USB) and the IEEE-1394 or *FireWire*[3] bus. Although they are aimed primarily at the consumer market, various controller and driver integrated circuits are available on the market that enable a hardware designer to realise such a bus on an embedded system. This, and the fact that they are both high performance buses, warrants their evaluation here.

Universal Serial Bus (USB)

USB is a peripheral bus standard developed by PC and telecom industry leaders that takes plug and play of computer peripherals outside the PC enclosure. The USB physical interconnect is a tiered star topology as shown in Figure 3.2. There are three components in any USB network: a host, functions or nodes, and hubs. There can be only one host in any network, and all wire segments are a point-to-point connection between the different components of the network. The final version of the USB specification 1.0 [Compaq et al, 1996] was released in 1995 and defines the architecture, communication protocol, device types, and type of connector. "At the end of 1999, the USB 2.0 specification will be officially released and will move the maximum transfer rate to between 120 and 240Mbps" [Wong, 1999:890].

The maximum bus speed on a USB version 1.0 network is 12Mbps. A host-scheduled, token-based protocol is used to address up to 127 peripheral devices in one network. The physical medium is implemented with a differential driver supporting bi-directional half-duplex operation over a maximum cable length of five metres between any two components. Error detection is implemented with a separate cyclic redundancy check

2 This term refers to those buses that usually interface between a standard IBM PC and other consumer products such as handheld video cameras, digital cameras, digital video disks (DVDs), video cassette recorders (VCRs), printers, palmtops, compact disk read-only memories (CDROMs) etc.

3 This is the trademark term used by *Apple Computer*, the original designer of the bus. In some Eastern block countries, IEEE-1394 is better known as i.Link, as used by the Sony corporation.

(CRC) for control and data fields. This provides 100% coverage for double-bit and less errors. It uses a four-wire connector with a single twisted pair for signalling, plus 5-volt power and ground conductors. USB supports asynchronous and isochronous [4] data transfer with maximum data lengths of 64 bytes and 1023 bytes respectively.

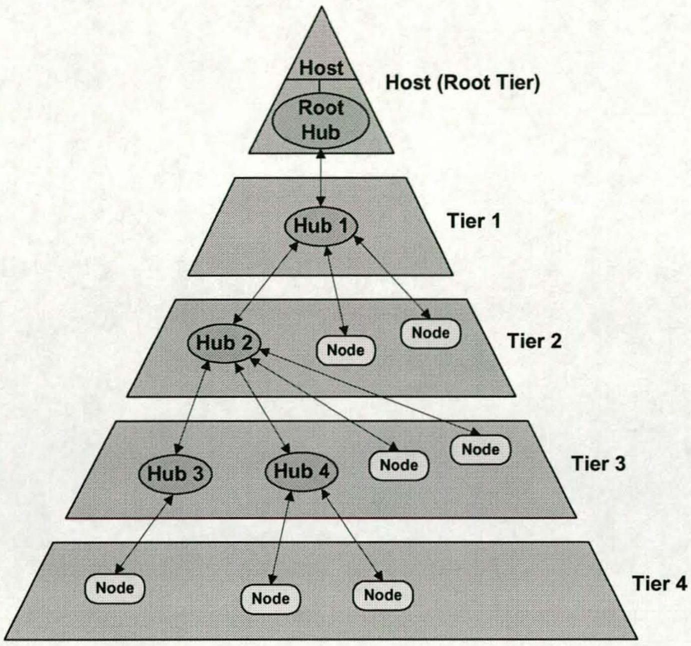

Figure 3.2 USB bus topology
(Taken from the USB specification document [Compaq, et al, 1996])

IEEE 1394 (*FireWire)*

The 1394 standard supports a peer-to-peer network with a point-to-point signalling environment, therefore no host is required. Nodes on the bus may have several ports - usually three. Each acts as a repeater, retransmitting any packets received by other ports within the node. The standard also defines two bus categories: backplane (12.5, 25 & 50Mbps) and cable (100, 200 & 400Mbps). The backplane bus is designed to supplement parallel bus structures by providing an alternate serial communication path between devices plugged into the backplane. A typical topology, mixing a backplane with a non-cyclic cable network, is shown in Figure 3.3.

A multi-master protocol supports both asynchronous and isochronous data transfer and also allows multi-speed transactions on the same bus, due to the point-to-point signalling

[4] The term refers to time dependent processes where data must be delivered within certain time constraints and in an uninterrupted manner. No error correction or retransmissions are available. Isochronous services are not as rigid as synchronous services, but are not as lenient as asynchronous services.

method. A clever data strobe encoding is used where either the data or the strobe signal changes in a bit time. The maximum data length is 512 bytes over a maximum distance of 4.5m between any two nodes. The cable carries both power (8V - 40V DC), and two signal pairs. A 64-bit address space makes provision for 1024 networks of 64 nodes each and 256TB memory space per node. The IEEE standard defines the mechanical interface as well as four protocol layers: physical, link, transaction, and serial bus management layers. Physical addresses are assigned on power up or bus reset, and whenever a node is added or removed from the system. No node ID switches are required, and hot plugging of nodes is supported.

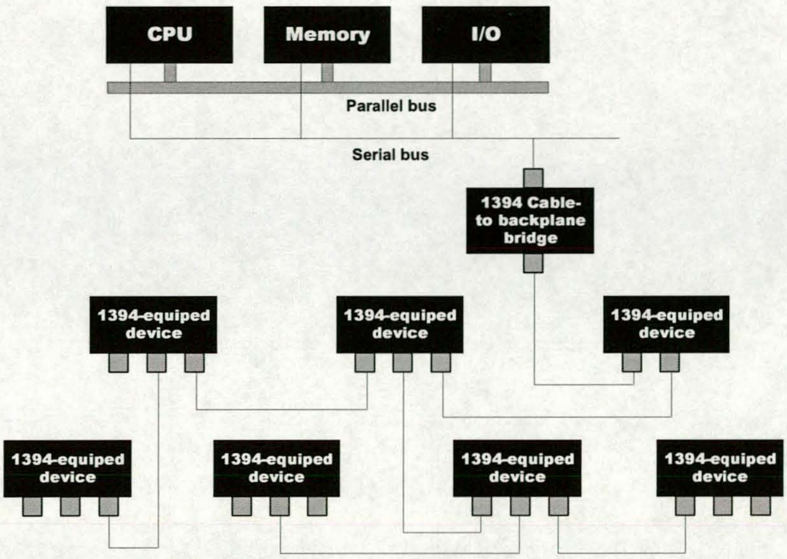

Figure 3.3 IEEE 1394 (*FireWire*) bus topology

Like USB, IEEE 1394 is also aimed at consumer products, but makes provision for much higher data transfer rates. According to preliminary data from market analysts, almost 100 million 1394-equipped consumer devices will ship in the year 2002 [Davoody, 1998]. The IEEE 1394 high-speed serial bus promises to revolutionise the transport of digital data for computers, and for professional and consumer electronics products. It "will provide" an inexpensive, non-proprietary method of interconnecting digital devices [Hoffman and Moore, 1995].

3.2.3 Fieldbuses

The set of serial buses aimed at general industrial automation are called fieldbuses. This broad definition comprises a wide spectrum of characteristics, all of which a system designer must take into account when planning a new design. Factors such as the type of application environment and ease of implementation, as well as the cost and availability of support tools and components, all play a significant role in the evaluation process. The following serve as applicable examples of popular fieldbuses:

Bitbus

This bus has been developed by Intel at the beginning of the 1980s as an open communication system and is optimised for the transmission of short real-time messages. Bitbus was officially standardised as IEEE-1118 in 1991. It has a master-slave structure where up to 28 slaves can be addressed per segment. Each slave has its own network address which makes it uniquely identifiable in the network. IEEE-1118 also makes provision for broadcasting and multicasting messages from the master, ie. sending messages to all or only a selected group of slaves.

The bit-rate of this bus is either 62.5Kbps or 375Kbps. It's total message length is 248 bytes. The physical medium is RS-485 with twisted-pair wire, and the Serial Data Link Control (SDLC) protocol is used to communicate between nodes. Bitbus can therefore be realised with any IC implementing the SDLC protocol stack. One of the reasons for the early success of Bitbus is the Remote Access Control (RAC) services. The RAC is a set of services for direct access to remote resources, tasks, I/O, memory, etc. In the ISO network reference model, the RAC functions cover the application layer, layer seven.

Profibus

Profibus is an international, open fieldbus standard developed in the late 1980s in Europe. It has since been standardised as EN50170 and EN50254. The Profibus family consists of three compatible versions: Profibus-DP for high-speed applications, Profibus-PA for process automation - mainly chemical applications - and Profibus-FMS for general-purpose applications. Both the DP and PA version implement the bottom two network layers, while the FMS version adds layer seven functionality as well. Profibus-FMS also permits data communication and power over the same bus according to international standard IEC1158-2. RS-485 is the transmission technology most frequently used by DP/FMS; and a unique transmission speed between 9.6Kbps and 12Mbps can be selected for all devices on the bus.

The bus access protocol used is a mixture of multi-master and master-slave: when the network is set up, certain nodes are designated masters and others slaves. A slave can only respond on the bus when asked to do so by a master. A master can send messages to slaves and other masters in a broadcast or multicast fashion only when it holds the bus access rights or token. The token is passed around between the masters in a logical ring. A maximum length of 224 bytes per message and up to 32 stations in each network segment is allowed. Figure 3.4 illustrates this process.

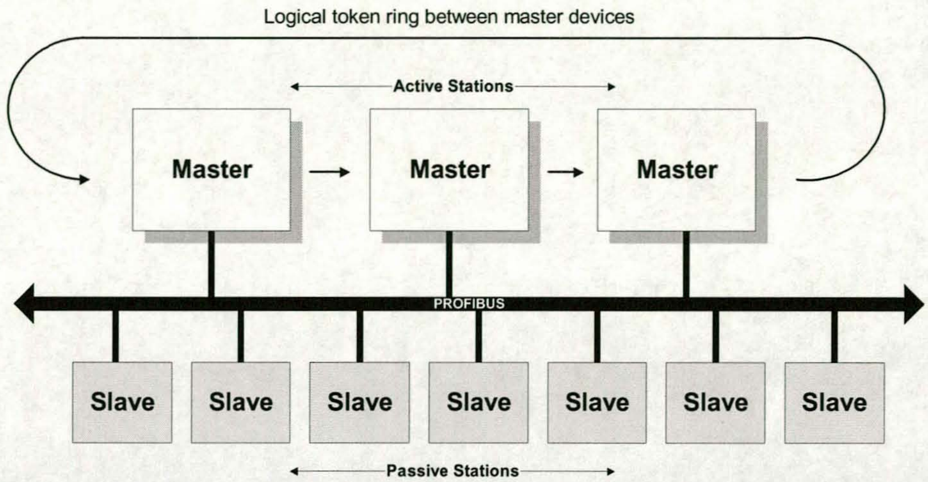

Figure 3.4 Topology of a Profibus network

LonWorks

LonWorks (Local Operating NetWork) technology was initially developed in the building automation sector by Echelon Corporation (USA) as a mechanism for intelligent devices to exchange control and status information. Public documentation is provided on the LonWorks protocol, called LonTalk. Lontalk is implemented as an integrated microcontroller and network communications device, the *Neuron*[5] chip. A modified version of the C programming language, called neuronC (ANSI-C plus three extensions) is used to program the *Neuron* chips. The protocol was designed to be independent of the communications medium used for the physical layer, and can be applied to twisted pair, fibre optic, power line modems and RF networks. It was also designed to support very large networks: up to 32 385 nodes in a domain and up to 2^{48} domains in a network.

The network topology is inherently peer-to-peer, but master-slave architectures can be supported if needed. LonTalk - implemented in hardware - provides services appropriate to a control network at all seven ISO network layers. It has been standardised as EIA-709.1 and is freely available for implementation on any microprocessor, fully interoperable with *Neuron* chip implementations [Rabbie, 1998]. LonWorks uses a modified version of Carrier Sense Multiple Access (CSMA) - called p-Persistent CSMA - as a media access

[5] *Neuron* chips are manufactured by Toshiba and Motorola but the latter announced its withdrawal from the *Neuron* market in January 1999. Each *Neuron* chip has a unique 48-bit ID. This is managed by Echelon Corporation at a cost of U$0.15 per node, incurred by the customer.

algorithm. The maximum transmission speed on a LonWorks network is 1.25Mbps and the maximum packet size, including data, addressing and protocol overhead, is 255 bytes. The data size within a packet is variable.

CAN

CAN (Controller Area Network) was originally designed by Bosch in 1985 in order to reduce wiring harnesses in the automobile industry [Lawrenz, 1997:3,26]. It has since been widely adopted in industrial applications worldwide and was standardised in 1994 as ISO 11898 for bus speeds above 125Kbps, and ISO 11519 for bus speeds below 125Kbps. CAN is a broadcast-type protocol implemented in hardware, that can be used to realise a multi-master network running at a maximum bus speed of 1Mbps over a maximum cable length of 40m. This length can be increased for lower bus speeds. The protocol defines the lower two (physical and link) layers of the ISO network reference model.

The bus access method used is CSMA/CD with an added way to resolve collisions: each message transmitted has a unique 29-bit ID[6] that is also used as a priority for that message. Messages with lower ID numbers have a higher priority. This means that when two messages are simultaneously transmitted, the one with the lower ID wins arbitration and continues to send the rest of the message. In this way, no time is lost in the arbitration process. Error detection is implemented with a 15-bit CRC, and error handling is achieved by retransmission. A CAN network is fault tolerant in that faulty nodes will stop transmitting messages after reaching a predefined error limit. Repeaters are needed for networks having more than 32 nodes[7], and the protocol limits the data portion of a message to 8 bytes. Since CAN only implements the lower two network layers, a higher layer protocol is needed to implement the specific application interface.

3.3 Choice and motivation for a preferred bus

Comparing bus systems is a very ambitious and ambiguous task because of the numerous parameters needed to specify a network, and the very complex interdependencies between the parameters. In this section, a brief comparison will be undertaken of the various buses presented above. It should be noted that the nature of this document does

[6] An 11-bit ID can also be used - see Appendix A.

[7] This number is dependent on the characteristics of the CAN driver used and may vary from one manufacturer to another.

not allow a rigorous comparison based on actual performance data as was for instance done by Crowder [1996] for a commercial aircraft application. Only four candidate fieldbuses out of many[8] were chosen as possible contenders for the new C&DH system due to their comparative and applicable characteristics. With reference to Table 3.1 at the end of this chapter and the preceding descriptions, Controller Area Network technology is chosen for the new C&DH system. The choice of CAN is based on general reasoning. This is outlined below and is followed by a presentation of the more specific considerations that are taken into account.

General comparison

Although the two proprietary SUNSAT buses are operational in space, neither of them implements sections of the ISO network layers in hardware. The software protocols therefore incur additional overhead. The master-slave architecture of the SSB makes the bus access restrictive. However, the bit-rate and number of bytes transmitted per message in each case are acceptable.

FireWire and USB are both elaborate protocols aimed at high-speed applications. Their maximum transmission rates are both at least two orders of magnitude higher than the lower-end specification for the new C&DH system (see next chapter). Both can be implemented in an embedded[9] environment though, although USB also needs a host. Any smaller scale USB design thus becomes overcomplicated. Full utilisation of the capabilities of the USB standard requires a version of the *Microsoft* operating system that supports USB at the operating system level. This renders it unsuitable for a satellite application. The start-up cost for a FireWire network is higher than that for USB, and the official specification document for the former can only be obtained from the IEEE at extra cost.

Of the four fieldbuses presented, only CAN has a truly multi-master capability, allowing considerable freedom in network design and operation. All have sufficient speed capabilities, and allow enough nodes per segment for a microsatellite application without the need for repeaters. LonWorks has a huge start-up cost compared to the others, and does not provide a very wide range of application ICs. Of all the buses presented, CAN has the lowest allowable data length per message - only 8 bytes.

---

[8] Other examples include Foundation Fieldbus, Sercos, Interbus-S, EiB, ASI, P-NET, WorldFIP, HART, Swiftnet, etc.

[9] USB device controllers include 8-bit derivatives like the Infinion C541U and the Zilog Z8E520.

In terms of hardware implementation of the fieldbus protocols, three out of four buses provide more than the bottom two layers of the ISO network structure. This negatively influences bus access times, since messages take longer to move up and down the protocol stack. All the commercial buses presented are aimed towards industrial or consumer applications. For space applications, a degree of freedom is therefore needed to add a customised application layer to the given protocol stack. This is offered only by Profibus and CAN.

In summary, it can be seen that the general reasoning above indicates the superior suitability of CAN as the most adequate bus architecture. More specific arguments in support of this decision are clarified below.

Pro-CAN

The CAN protocol is implemented on a wide range of ICs, including stand-alone protocol controllers, and integrated microprocessor controllers on various platforms. The latter includes the popular Intel 8-bit MCS-51 architecture already widely used throughout SUNSAT-1. In addition, various vendors offer development and diagnostic tools at low cost that significantly ease and speed up the implementation of a 'first network'. Thirdly, the protocol is not complex, and the bus specification documentation is publicly available at no extra cost.

CAN is a robust protocol and has several error detection and fault tolerant capabilities, rendering it suitable for space applications. It has been repeatedly proven in critical applications such as the ABS breaking and engine control systems found in luxury sedans. The multimaster node hierarchy ensures that the whole network does not collapse when a single node fails - the system performance is only gracefully degraded. As a result, true event driven communication is possible as needed for setting TCMDs and routing TLM.

However, CAN is not suitable for the transfer of large amounts of data. In the C&DH application presented, the 8-byte data length is expected to be ideal for TCMD setting and the transfer of A/D samples within a satellite at relatively high speed. Either all nodes in a CAN network receive the same information at the same time or none at all. Data consistency is thus ensured throughout the network. If at least one node detects an error while receiving a message, it will signal an error frame. The error frame causes a retransmission from the transmitting node, and it signals to all other nodes to discard the recently received message.

CAN is known as an autobus protocol since it was designed for that industry and today still accounts for the majority of all CAN devices being sold worldwide[10]. So the question might be asked: "What about other[11] autobus protocols"? The answer lies in the fact that none of the others have also migrated towards industrial applications as CAN has. Furthermore, VAN and ABUS have already been abandoned in favour of CAN. All have more or less the same characteristics, but only CAN offers higher speed communications (above 500Kbps). Thus, it can be concluded that CAN has been proven in both automotive and industrial designs as a high performance, low cost solution to control applications with multiple sources in terms of semiconductor suppliers [Lawrenz, 1997:26]. Additional background knowledge about CAN may be found in Appendix A.

3.4 Conclusions

This chapter evaluated eight different serial buses with the view to select one for inclusion on the next generation C&DH system. Two buses currently used on SUNSAT-1 were found to be too restrictive in terms of architecture and SW overhead. *FireWire* and USB do satisfy the requirements of the proposed C&DH system, but would overcomplicate it due to their intricate architecture and elaborate protocols. From the evaluation of four fieldbus architectures, it was shown that CAN provides a simple and low cost, though fault tolerant and reliable vehicle for relatively high speed (up to 1Mbps) communication over a single twisted-wire pair.

[10] Over 9 million CAN interfaces have been sold between 1989 and 1997 [Lawrenz, 1997:29] and the expected total sales for the year 2000 approaches the 140 million mark [Electronics World, 1999:888].

[11] For instance VAN, J1850, ABUS, CCD and PALNET.

Table 3.1 Comparison of serial buses

Bus type	Year Introduced	Governing standard	Openness	Physical media	Maximum nodes	Distance @ max bitrate	Maximum bitrate	Communic. method	Data transfer size	Arbitration method	HW ISO network layers	Manufacturers	Startup cost
SSB	1994	none	closed proprietary bus	signal plus ground	15	100m	19.2Kbps	Master-Slave	250 bytes	none	none	-	< R100
IBUS	1994	none	closed proprietary bus	2-wire twisted pair	32	500m	19.2Kbps	Multi-Master	31 bytes	CSMA/CD	none	-	< R100
USB	1995 (spec.)	Industry standard	open standard	2 twisted pairs	127	Max. 5m between 2 nodes	12Mbps	Host-scheduled Token-based	64 Asynchronous 1024 Isochronous	none	non-conforming	Mainly Intel but development tools and drivers from others	±R1000
FireWire	1995 (spec.)	IEEE-1394	open standard	3 twisted pairs	64 per network	Max 4.5m between 2 nodes	50Mbps 400 Mbps Back, Cab	Multi-Master Peer-to-Peer	512 bytes	custom	Physical Link Transaction	ICs from Texas Instruments and others + many consumer products	±R1000 to R7000
BitBus	1991 (spec.)	IEEE-1118	open standard	twisted pair	28	300m	375Kbps	Master-Slave	248 bytes	none	1, 2 and 7	Intel, Motorola, Toshiba and others	No info available
Profibus	Late 1980's	EN50170 EN50254	open standard	twisted pair or fibre	32 per segment	100m between segments	12Mbps	Multi-Master Peer-to-Peer	224 bytes	Masters pass a token	1 & 2 + 7 for FMS	Siemens, Profichip. Products from over 300 vendors	No info available
LonWorks	1991	EIA-709.1	Public documents on protocol	twisted pair, fibre RF power	32 000 per domain	300m	1.25Mbps	MasterSlave Peer-to-Peer	228 bytes	CSMA	1 to 7	Toshiba. Motorola withdrew from market	±R20 000 to R120 000
CAN	1994 (spec)	ISO11898 ISO11519	open standard	twisted pair	32	40m	1Mbps	Multi-Master	8	CSMA/CD and collision resolution	1, 2	Siemens, Intel, Philips and others	< R1000

Chapter 4

Structural design of the new C&DH system

CAN technology will be used to implement a serial bus for the new C&DH system. This chapter evaluates various system configurations for this bus, and where it fits into the proposed electronic design of the rest of SUNSAT-2. It is, however, important to first establish the guidelines and preliminary specifications for the proposed design.

4.1 Design guidelines

Two types of design guidelines, a design methodology and a short list of preliminary specifications are provided below before looking at the proposed design in more detail.

4.1.1 Design methodology

Probably one of the most important design requirements on any spacecraft, is **redundancy**, or "the addition of information, resources, or time beyond what is needed for normal system operation" [Johnson, 1989:49], including the now standard duplication of hardware. **Fault tolerance** stems from the above and is defined as the design of systems that can continue the correct performance of specified tasks in the presence of hardware failures and software errors [Johnson, 1989:39]. Together, these two aspects form the foundation for a reliable design.

As pointed out by Jilla [1997], **physical modularity** and electronic flexibility can also greatly contribute to a more reliable system. The first allows new technologies to be incorporated into the basic design more easily. On the other hand, **electronic flexibility** allows the ground controllers to choose between various satellite configurations depending on the performance of the satellite's subsystems. For increased reliability, some paths include only flight-proven technology while other paths use new technology for increased performance. It can be seen that physical modularity pays off on the ground, and electronic flexibility is its in-orbit counterpart.

4.1.2 Preliminary engineering specifications

The list of specifications below is still incomplete since it depends on the specific interfaces between the proposed C&DH system and other subsystems on the new satellite and also on the details of the satellite bus[1]. Both of these aspects still need to be finalised. Preliminary specifications related to the new C&DH system include, but are not limited to:

- Minimum sustained transmission rate[2]: 96Kbps
- Minimum digital TCMD lines per subsystem: 20
- Minimum analog TLM channels per subsystem: 20
- Analog-to-digital conversion accuracy: 10 bits
- Available TLM bandwidth should be optimised. The TLM system should occupy bandwidth only when necessary.
- The states of all TCMDs should be known (both on-board and on the ground) at any given stage.
- The joint TLM and TCMD system should be able to function independently from the on-board computers, and should be operational at all times.
- The interface between the C&DH system and other subsystems on the satellite should be standardised to facilitate ease of implementation.
- Commercial off-the-shelf (COTS) components - available in an extended temperature range - should be used in the implementation of the C&DH system[3].
- The C&DH system should be able to enter a reduced power consumption mode during times in orbit when its functions are not required.
- The TLM system should be able to dwell on specific channels when needed.
- The C&DH system should be able to initialise all TCMDs to a known state via a system-wide reset.
- No single point of failure should permit a command latch to become non-functional.
- The TCMD system should be able to accept asynchronous commands from the modems (via the ground station), the OBCs and other subsystems.
- The data formats and transmission protocols on the C&DH system should comply with current international standards.

---

1 'Bus' in this context refers to the overall physical (mechanical) structure and layout of the different subsystems within the satellite.

2 This constitutes an order of magnitude improvement over the maximum TLM sampling rate on SUNSAT-1 (9600-Baud). Transmission in this sense means communication between the different components of the C&DH system, and between the C&DH system and the on-board computers.

3 COTS components ease availability and adapt more easily to budgetary constraints compared to MIL-spec components. "A cost saving and [technical] performance improvement is generally associated with commercial devices but must be offset in comparison to the potential for increased costs associated with their decreased [environmental] performance specification" [Day, 1999]

4.2 The proposed SUNSAT-2 satellite bus architecture

The electronic and mechanical design of SUNSAT-2 remain at a very early stage of planning. As a result, only preliminary thoughts surrounding system layout and interaction have been discussed. The ideas presented here are an extension of these discussions.

4.2.1 Tradeoffs between centralised and distributed architectures

The centralised system design of the TCMD, and in particular the TLM system on SUNSAT-1, has certain limitations. Firstly, the design of these systems is inflexible, which makes it difficult to apply and adapt them for future satellite designs. Furthermore, the fact that control has to be distributed from, and data has to be gathered at a central point, complicates and enlarges the cable harness. Finally, centralised designs inherently pose problems when critical components fail. The performance of the whole system can be severely degraded in such a case. With the SUNSAT-1 TLM and TCMD systems, this problem was countered through the duplication of many of the components and sections of the designs. This again increased the physical board space available.

A distributed architecture offers benefits in the following areas:

- Facilitation of design for multiple missions instead of a unique mission. This reduces development time and implementation costs.
- The risks of system failure are spread throughout the satellite, instead of being concentrated at a single point. This does not imply that redundancy should be abandoned, but it can be scaled down to limit board space and cost.
- A distributed architecture leads to modular design, simplifying system integration and testing.

The lower part of Figure 4.1 shows how the former centralised C&DH system (combined TLM and TCMD subsystems) can be split into smaller units called nodes. Each node has the capability of issuing TCMDs and of gathering TLM. CAN lends itself perfectly for the implementation of a distributed architecture which is proposed here for the new C&DH system.

4.2.2 A dedicated bus for C&DH

Previous work [Koekemoer and Bakkes, 1999] proposed that the new C&DH system be based on a dedicated bus as shown in Figure 4.1. The main advantages of such an

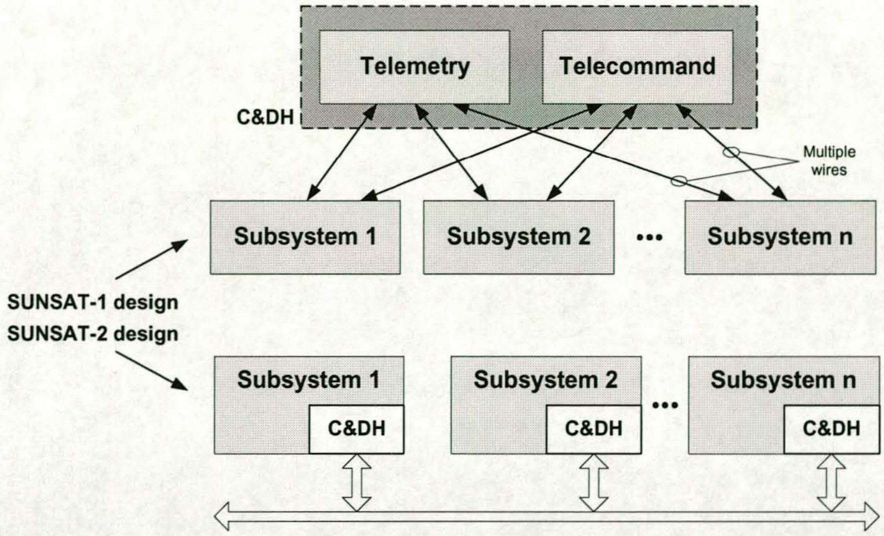

Figure 4.1 Comparison between the SUNSAT-1 and proposed SUNSAT-2 C&DH designs

implementation are higher reliability and less dependability on the resources of the on-board computers. It also makes the C&DH much more expandable: additional payloads and subsystems can be added without having to change the architecture of the design - nodes are simply added to the existing bus. For instance, only four sample subsystems are shown in the more detailed Figure 4.2, but by maintaining the same level of C&DH complexity, many more can be added.

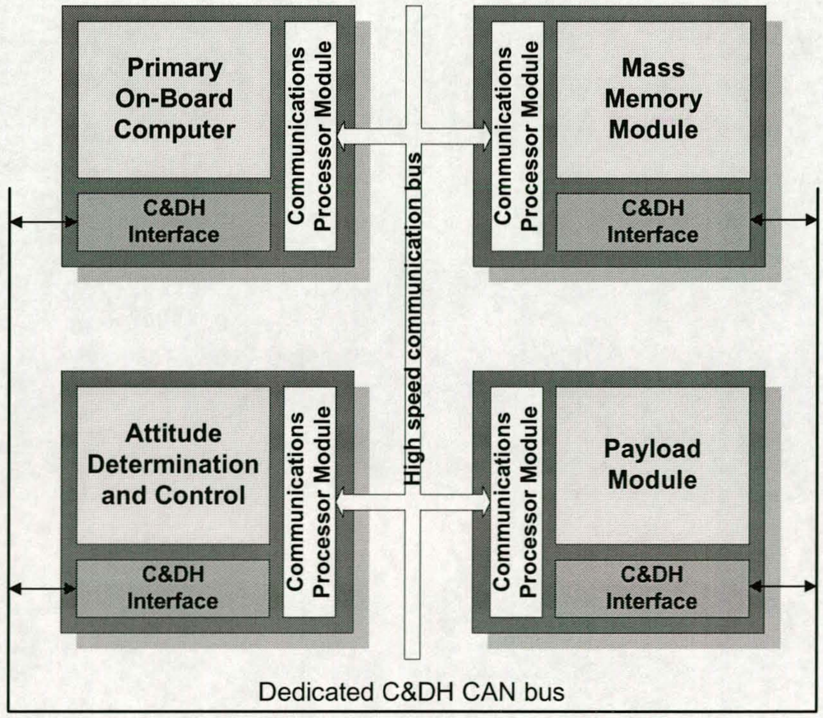

Figure 4.2 SUNSAT-2 bus architecture

The function of a communication processor module is to act as an interface between a subsystem and one or more higher speed buses for general data transfer. These modules should also be designed as modular units, so that they can be employed on all subsystems with little modification. Using this design methodology, even a subsystem becomes a modular unit, with the core function being application-specific.

4.2.3 Generic subsystem design

The proposed make-up of a sample subsystem is shown in Figure 4.3. In this case, the primary OBC is used as an example. The C&DH unit, or node, will issue TCMDs directly to the subsystem core, as well as to the communications module. It is also responsible for gathering TLM from both these entities. It communicates directly with other C&DH nodes via the dedicated C&DH bus, shown at the bottom of the figure.

An interface between the C&DH unit and the communications module enables the latter to set TCMDs on the subsystem core, in the event that the interface between the C&DH unit and C&DH bus becomes dysfunctional. Three methods exist for setting TCMDs on a subsystem:

- A command to do so is received via the C&DH bus;
- The communications module sets the TCMD directly via the C&DH node; and
- An intelligent source like a micro processor or FPGA on the subsystem core sets the TCMD itself.

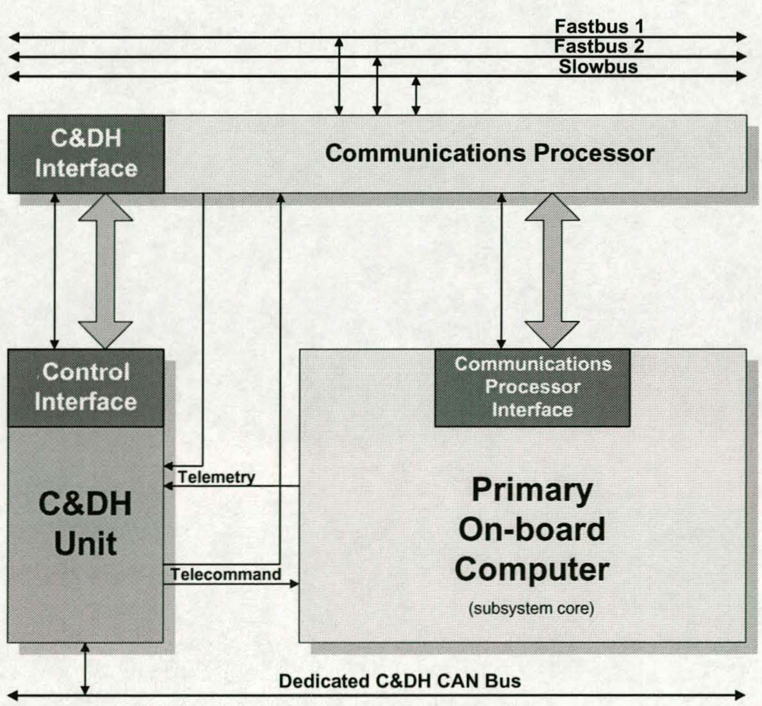

Figure 4.3 A SUNSAT-2 generic subsystem

Delayed TCMDs are issued by the C&DH node when a system clock on the node reaches a predetermined value. This clock can be maintained via the C&DH bus or communications module for instance. The state of local TCMD switches on the subsystem core can be read back by the C&DH node and made known to other subsystems or C&DH nodes via the C&DH bus. TLM acquisition, however, is contained within each node and cannot depend on system-wide redundancy as back-up. The detail of the communication buses shown at the top of Figure 4.3 is not important at this stage, but may involve one or more fast and/or slow buses depending on the payload and house-keeping needs of SUNSAT-2.

4.3 CAN and the new C&DH bus

Over the past 15 years of CAN existence, several integrated circuits have been developed to meet the different needs of both automotive and industrial applications. This leads to some interesting design scenarios, presented in this section.

4.3.1 Implementation options based on available devices

Today, most large semiconductor companies[4] around the world offer CAN devices in various configurations. Two distinct groups of components exist: CAN as an on-chip peripheral on various microcontroller architectures; and stand-alone components that implement only the CAN protocol in hardware. The microcontrollers include the popular Intel 8-bit (MCS®51) and 16-bit (MCS®96) architectures. For both these groups, a separate transceiver is needed between the CAN device and the CAN network.

Figure 4.4 shows the possible node configurations available for implementing CAN. Some companies[5] recently started to make CAN controller cores available, which can be implemented on an FPGA as shown in the fourth group in Figure 4.4. These cores remain very expensive, but are ideal for critical space applications where they can be synthesised in radiation-hardened or radiation-tolerant FPGAs. Devices for the third group have only recently become available[6]. A fifth group of components, not shown in the figure, is Serial Link Input Output (SLIO) devices which are a reduced version of group 2. They cannot act as masters in a CAN network, and whenever a CAN message is received by such a device, the corresponding data is directly written into the related output port. These low-

4 For instance Intel, Infinion, Philips, Motorola, Texas Instruments etc.

5 For instance Inicore and Atmel.

6 For instance the Philips P8xC592

cost devices are not very flexible and thus not very popular.

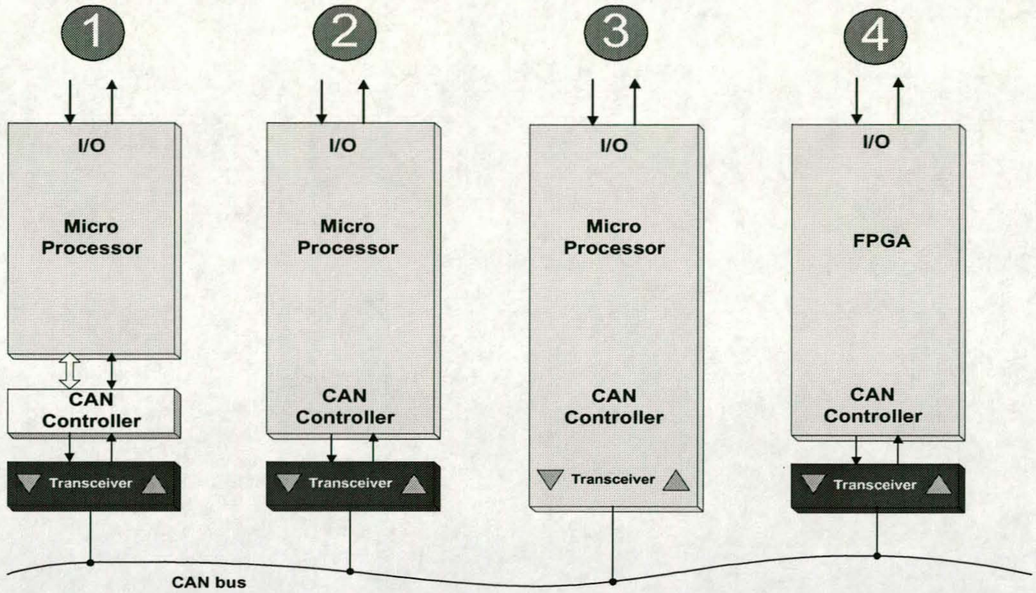

Figure 4.4 CAN node implementation options

When deciding on a specific implementation, four main factors must be taken into account: **Implementation cost:** with integrated CAN, development cost is lower due to a smaller size printed circuit board. Manufacturing cost is also lower, since in high volumes, semiconductor manufacturers can offer CPUs with integrated CAN for a lower price than separate CPUs and CAN chips.

Design flexibility: software developed for an integrated CAN peripheral of one CPU may not apply to a second CPU with on-chip CAN, especially if the CPUs are supplied by different vendors. In this regard, stand-alone devices offer more flexibility.

Level of CPU burden: the level of CPU burden to maintain on-chip CAN is approximately half the burden of a stand-alone CAN device [Szydlowski, 1994]. Protocol tasks such as bus arbitration and CRC calculations are independent from the CPU, but messaging tasks are not. A CPU with on-chip CAN will read/write to register locations using an internal bus, but will have to use the external address/data bus to do so with a stand-alone device. However, it should be noted that some micro controllers treat the internal CAN memory area as an 'external' address space, using *MOVX* instructions to manipulate the various registers inside the memory. In such a case, there is also no difference in CPU burden between integrated and stand-alone CAN devices.

System reliability: integrated CAN peripherals have a reliability advantage over stand-alone devices because of their smaller form factor, and since the integrated peripheral is burned-in and tested together with the CPU. Less printed circuit tracks, and solder joints in the case of the former, are also conducive for reliability. In addition, integrated CAN applications simplify hardware design, generate less board noise, and are easier to test and repair.

The tradeoff between integrated and stand-alone devices may thus be viewed as the value of design flexibility versus implementation cost, level of CPU burden and system reliability [Szydlowski, 1994]. In the satellite application at hand, the CAN node implementation itself offers enough flexibility already. This means that the other three factors are the motivation for choosing integrated peripheral devices for the new C&DH system.

4.3.2 Redundancy issues

CAN is a fault-tolerant bus architecture with built-in error detection capabilities and also offers fault-tolerant transceivers. For industrial and automotive applications, this might prove to be sufficient for secure applications; but in a harsh, distant space environment, redundancy measures are necessary to guarantee successful operation. This section by no means presents an in-depth discussion of this complex field of study but merely looks at some of the possibilities available to implement redundancy with CAN in the new C&DH system on three different levels. These possibilities are therefore an extension of the redundancy design methodology presented earlier.

Channel-level redundancy

The physical communication channel is a critical part of any distributed system: in the worst case, when it suffers a permanent failure, the operation of the whole system can be jeopardised. A double communication channel is therefore proposed for the C&DH system. This is illustrated in Figure 4.5. Tolerance of permanent failures in any part of a specific channel is thus allowed; and can be implemented using only one transceiver for both channels with a switch-over technique or a transceiver per channel (see below). Special fault-tolerant transceivers[7] have also been developed. They can tolerate various fault conditions on the communication channel. However, most of these transceivers limit the bus speed to 125Kbps.

[7] Such as the Philips 82C252 and the Infinion TLE6252

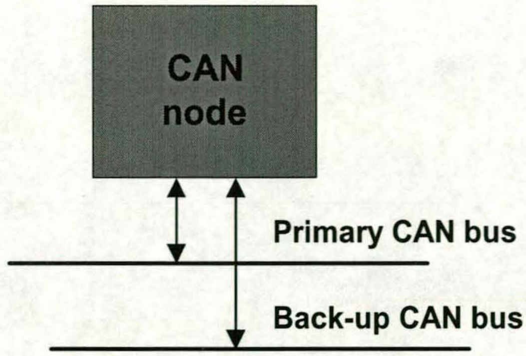

Figure 4.5 A double-CAN bus

Component-level redundancy

This is a form of spacial redundancy where certain components are duplicated in order to increase the reliability of the system. Examples are the two transceivers mentioned above. New advances in CAN devices offer other interesting possibilities. There is a recent trend toward manufacturing integrated CAN microprocessors with more than one built-in CAN controller[8]. This results in a very wide range of applications, especially as gateways in inter-network communications [Barrenscheen, 1998]. These devices offer a form of component duplication, but avert the drawback of increased component count and thus increased board space. Such devices can also be used in conjunction with the double communication channel presented above. Two possibilities are shown in Figures 4.6 and 4.7 respectively [Wolf and Koller, 1998].

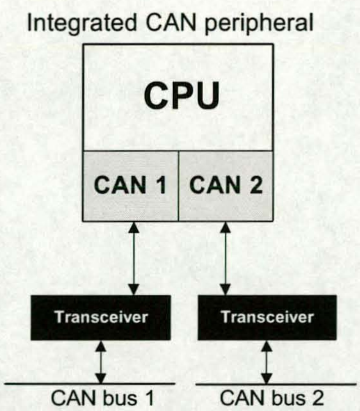

Figure 4.6 TwinCAN implementation 1

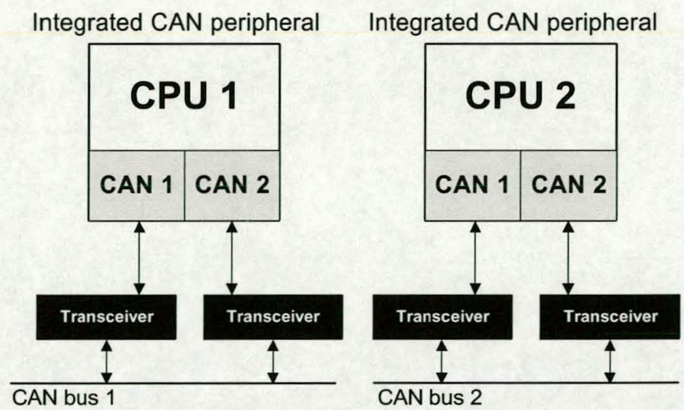

Figure 4.7 TwinCAN implementation 2

One processor is used to address two different communication channels that can run at different speeds in Figure 4.6. A failure in any of the transceivers will leave the corresponding bus inaccessible to the processor. An improvement is shown in the next figure, where each bus is addressed by a separate processor. Not only does this leave

[8] Such as the 'TwinCAN' module from Infinion (C167CS) and the 'Atomic' module from NEC (based on the V850 CPU core)

room for transceiver failure, but also for partial - where only one CAN controller in the microprocessor becomes faulty - and full microprocessor failure. For each bus, the processor compares the outputs of both CAN controllers as a measure of redundancy. With this implementation, a decision point will be needed to arbitrate between the two processors as is proposed by Ferriol et al [1998]. Clearly, the solution presented in Figure 4.7 offers improved reliability, but it also raises the complexity of the system.

System-level redundancy

This form of redundancy stems from the natural structure of a communication system consisting of a set of interconnected nodes. In this scenario, it is possible for a node to assume the functions of another node in case of failure. The multi-master capabilities of CAN are tailored for this type of behaviour. With the proposed C&DH architecture, system-level redundancy is accomplished within the communication processor module on each subsystem. When a certain C&DH node fails, a neighbouring node may receive an instruction to set a TCMD on the subsystem that contains the faulty C&DH node. This is done via its communication processor module and the CAN interface of the neighbouring node. In addition, the CAN protocol itself provides information redundancy in the form of a 15-bit CRC word inserted at the end of each transmitted frame. Time redundancy is provided in the form of automatic retransmission, when errors on the bus or in a message have been detected.

4.3.3 C&DH node detail

This paragraph uses the ideas presented in the previous sections to focus on the specific architecture of the C&DH nodes. The implementation details thereof follow in the next chapter.

A high-level diagram of the proposed C&DH node implementation is shown in Figure 4.8. It contains a microprocessor with a built-in CAN interface, communicating with a dual-redundant CAN bus via two separate CAN transceivers. Note that the structure shown in Figure 4.8 is similar to the *TwinCan* implementation of Figure 4.6; but in this case a micro processor with only one CAN interface is used. To allow the processor access to CAN bus 1 in the case when the transceiver for that bus fails, a transceiver switch-over technique must be implemented.

An A/D is used to convert the TLM samples into a 10-bit digital format. All TCMDs will be issued by the processor itself, and this will also monitor the status of the TCMDs directly.

With reference to Figure 2.1, the micro processor in Figure 4.8 becomes the command decoder, and the sources of the commands are an on-board computer (via the communication processor module or the CAN bus), an RF link (also via the CAN bus) or a test port. Three test ports are indicated in Figure 4.8, one each for the two CAN buses and another that connects to the processor via the serial port. The two test ports on the CAN buses need not be physically located on a particular CAN node. For practical purposes they can be located anywhere on the CAN bus. TLM processing (as shown in Figure 2.2) is handled by the micro processor which can also format, compress and store the TLM, or send it to a physically separate storage area, such as a RAMDISK, via the CAN bus.

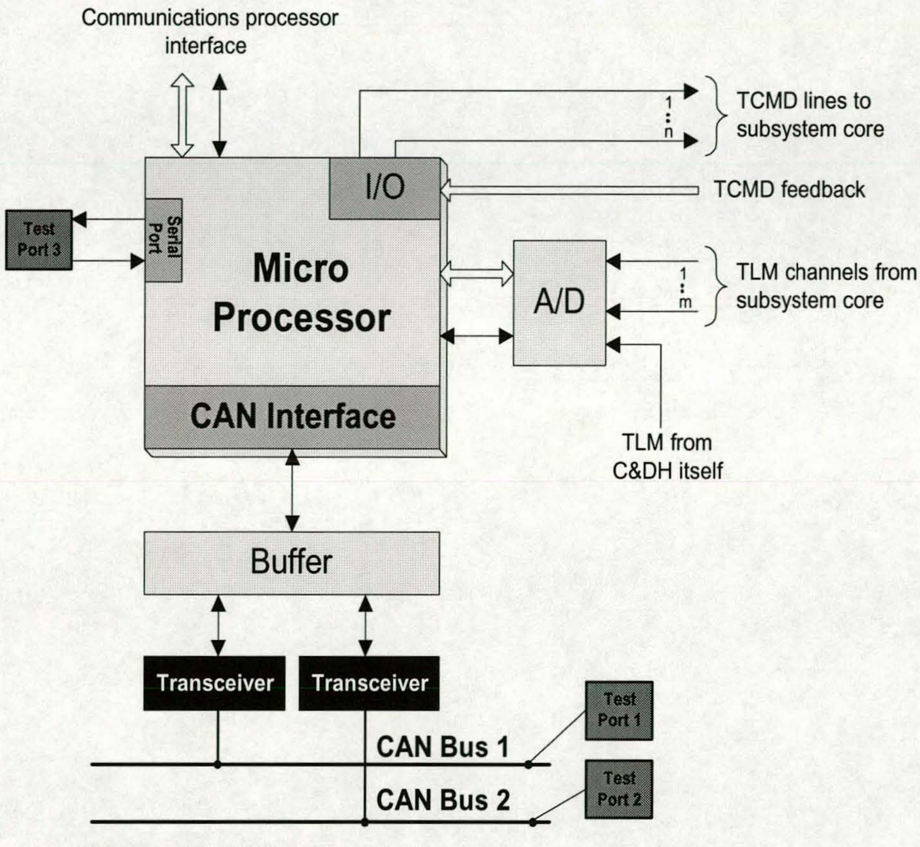

Figure 4.8 CAN node detail

4.4 Conclusions

The structural design of the new C&DH system presented in this chapter conforms to the principles of electronic flexibility and physical modularity. It becomes very easy to add more nodes if payload requirements should change. The function of a node can also be extended when other nodes become faulty. The importance of fault tolerance is addressed by basing the design on a dual-redundant CAN network, and by making use of component-level redundancy coupled with the built-in fault tolerant capabilities of CAN.

Furthermore, a degree of satellite autonomy is realised by implementing a micro processor on each node. This further extends the functional capabilities of the satellite, especially in remote parts of orbit.

In Chapter 2 a packetised C&DH system was shown to be more advantageous than a TDM system. An inherently packetised system was realised with the implementation presented in this chapter: each C&DH node packs the TLM data into a CAN frame, and then transmits it over the CAN bus. In a similar way, TCMDs are received over the CAN bus and the data is then extracted from the CAN frame (TCMD packet).

Chapter 5

Implementation of the test setup

In this chapter, the design details of the test C&DH node are presented. This is followed by a description of the memory mapping of the node; and a brief timing analysis. It is shown that the test setup meets the majority of the design specifications laid out in the previous chapter. Appendix B contains the detail schematic of the implementation presented here.

5.1 C&DH node components

The design of the C&DH node prototype can be divided into three system components, shown in Figure 5.1. It involves a computer system and data and communication interfaces. The first of these is the controlling component and interfaces with the other two components to control the global system configuration. The communication interface contains the connections to the CAN network, while the data interfaces contain the TLM and TCMD specific implementations along with a connection to a serial peripheral device like a PC.

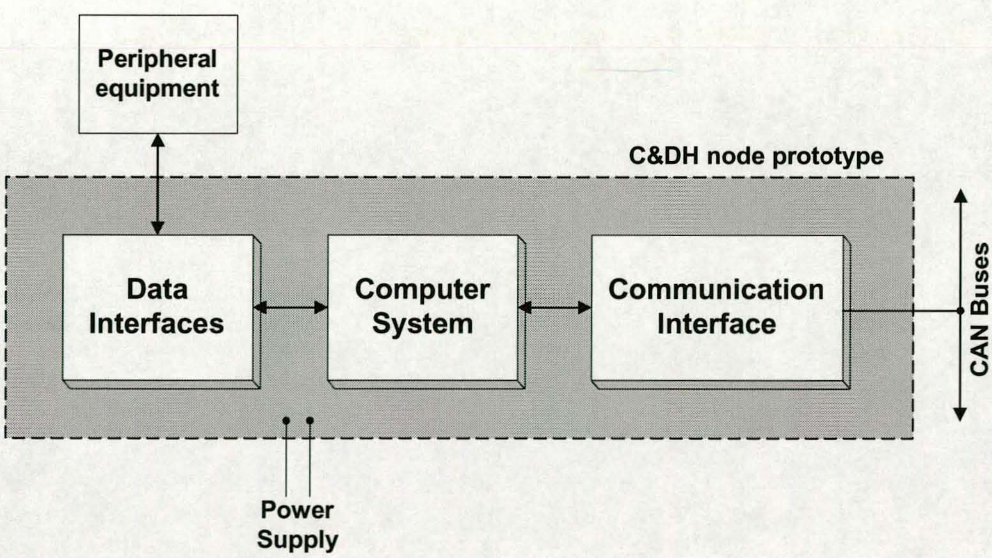

Figure 5.1 C&DH node hardware design

5.1.1 Computer system

The core components of the computer system are shown in Figure 5.2. These include a central processing unit (CPU); program memory; data memory and a CPLD. The selection and details of these are discussed below.

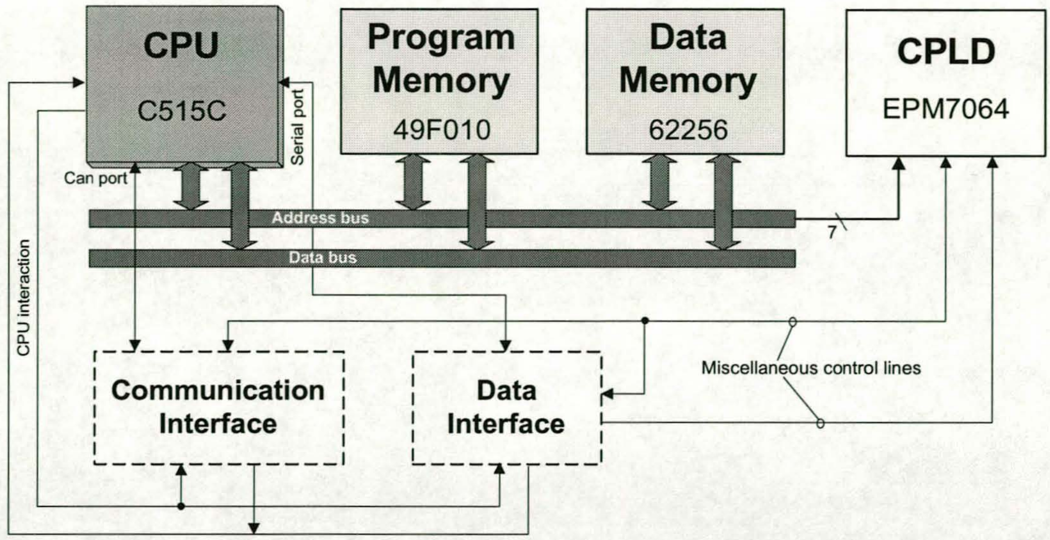

Figure 5.2 C&DH node computer system

CPU - The Infinion C515C microcontroller

The SAB-C515C-LM 8-bit CPU with built-in CAN from Infinion has the following main features:

- Intel MSC®51 class architecture with 8 datapointers;
- 256 bytes on-chip RAM;
- 2KB on-chip XRAM;
- 64KB external program and data memory;
- Up to 10MHz external operating frequency;
- Eight ports: 48+1 digital I/O lines, 8 analog inputs;
- Three 16-bit timers/counters;
- 10-bit A/D converter with multiplexed inputs and built-in self calibration;
- Full duplex serial interface (USART) with programmable baudrate generator;
- Synchronous Serial Channel (SSC) - SPI compatible;
- 17 interrupt vectors at four priority levels selectable;
- Extended watchdog facilities; and
- Power saving modes (slow-down, idle, SW power down, HW power down).

Various other CPUs[1] with built-in CAN are available on the market, but the Infinion derivative was chosen for its specific set of built-in peripheral devices, and also because

[1] For instance, the 8X196CA (Intel), 68HC08ZA (Motorola), P83CE598 (Philips), PIC18C241 (Microchip)

many vendors offer a range of inexpensive development tools and starter kits[2] for this processor. The built-in A/D converter will be used to sample TLM in the place of using an external converter as shown in Figure 4.8. In comparison with the C505C from Infinion, the C515C offers a SSC interface; 10-bit A/D conversion accuracy instead of 8-bit accuracy; more internal XRAM; more I/O lines, and consumes less power due to a lower maximum clock speed. The 16-bit Infinion derivatives with CAN have built-in program flash memory, but consume much more power - in the order of 100mA compared to about 19mA of the C515C for the same clock speed. In addition, the processing and memory needs of the C&DH nodes do not justify the use of a more powerful 16-bit processor. A new TwinCan device[3], suitable for the implementation shown in Figure 4.6, was only released during the third quarter of 1999 and could therefore not be evaluated as part of the work for this thesis.

Program memory - The Atmel AT49F010 flash memory

Flash memory is chosen so that the system node software can be updated via the CAN network during operation. This ability further satisfies the flexibility requirement for the new C&DH nodes. Since program code cannot be executed from the flash device when it enters a programming cycle, a piece of the program code has to be copied over to another memory component prior to programming the flash. The section on memory mapping below discusses this issue in more detail.

A AT49F010-70PC flash is used here and has the following features:

- 1 Mbit (128KB x 8)[4] memory;
- Single-voltage operation: 5V read and 5V programming;
- 8KB bottom boot block with lockout feature;
- Low operating power consumption (30mA maximum);
- Low stand-by current (100μA maximum); and

---

2 For instance, the kitCON-515C from Phytec corporation.

3 The Dallas Semiconductor device, DS80C390, is based on the 8051 micro controller core and has two separate CAN controllers, each with 15 message buffers. In addition, it also offers a mathematics accelerator, 4KB of on-chip SRAM, a 40MHz crystal speed, can address 4MB of external data and program memory, and only consumes 35mA at a clock speed of 40MHz (all pins disconnected).

4 The C515C only has 64KB of addressable memory space, but the 128KB flash device is used here primarily due to availability. The complete SW suit for a C&DH node will dictate the final memory requirements, but it is envisaged that 64KB should suffice. In future, an even larger flash memory with separate erasable blocks can be used to accommodate two sets of program SW: one for the original set of system SW, and the other to evaluate a new set of uploaded SW.

- Hardware data protection.

The memory map of the 49F010 is shown in Figure 5.3. It contains only one continuous block of memory, excluding the separate boot block, which can be erased in bulk. This specific device is chosen because it is one of the few 5V-only flash devices that is also available in a DIP package. For initial system testing, it is convenient to be able to remove the device for reprogramming in an universal programmer. The bottom boot block is also expedient when it comes to reprogramming the main program memory. The device requires only 5V for programming, which eliminates the need for an extra programming supply voltage on the board. When the CPU enters a low-power mode, the flash memory will be switched automatically into a stand-by mode.

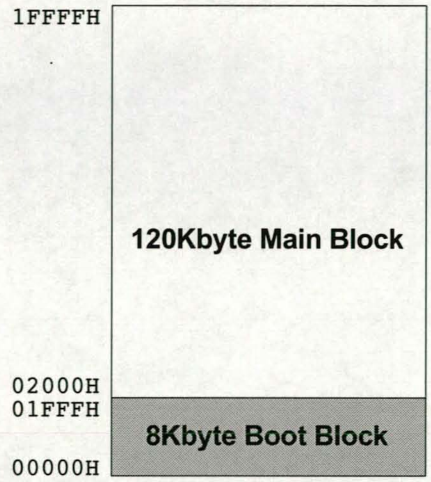

Figure 5.3 Flash memory map

Data memory - The Hitachi 62265 SRAM

The function of the data memory is to temporarily store data received through the user interfaces, and also to store a new set of program code before it is programmed into the flash memory. Program code can also be executed from this device. A HM62256BLP static RAM is used with the following features:

- 32KB 8-bit memory
- Low operating current (33mA maximum)
- Low standby current (0.2μA typical)

There are very few 64KB SRAM devices to match the 64KB address space of the CPU available on the market. This 32KB device was thus chosen as it is sufficient for development purposes. The main advantage of using SRAM is its low power consumption, and the ease with which it can be implemented. The C515C controls the chip select line of the SRAM, and can disable the chip for power-down mode.

CPLD - The Altera EPM7064

The CPLD is used in this design to implement 'glue-logic' and address decoding functions, chip select signals, and a digital multiplexer (see data interfaces section below). The EPM7064S-10 device is chosen due to the user-friendly development environment[5] available from Altera - as compared to that of the Xilinx corporation - and its ease of use. Provision is made on the C&DH development board for an ISP port, so that the CPLD can be programmed in-system. The device has the following features:

- 1250 usable gates
- 64 macro cells
- 4 logic array blocks
- 36 user I/O pins

To limit board space, and since no more than 36 I/O pins are needed, the 44-pin PLCC version of the device is used.

5.1.2 Data interfaces

The outlay of the data interfaces is shown in Figure 5.4, and consists of a RS-232 transceiver, a TCMD, and a TLM emulation section. These three components are discussed in more detail below.

RS-232 transceiver - The Maxim MAX232

An industry-standard RS-232 transceiver (MAX232CWE) is used to facilitate communication between the test C&DH node and a PC. In this way, it becomes easier to debug development SW and the transceiver can also be used to upload new program code to the flash memory. When the full C&DH node SW is completed, the RS-232 interface can be used as a test port for system verification or during integration. One transceiver of this dual channel device is directly connected to the serial port of the C515C.

TCMD emulation

Figure 5.5 shows the proposed implementation of a TCMD switch. The output of each gate is fed back to the C&DH node where the current state of the switch is then determined.

---

[5] MaxPlus-II

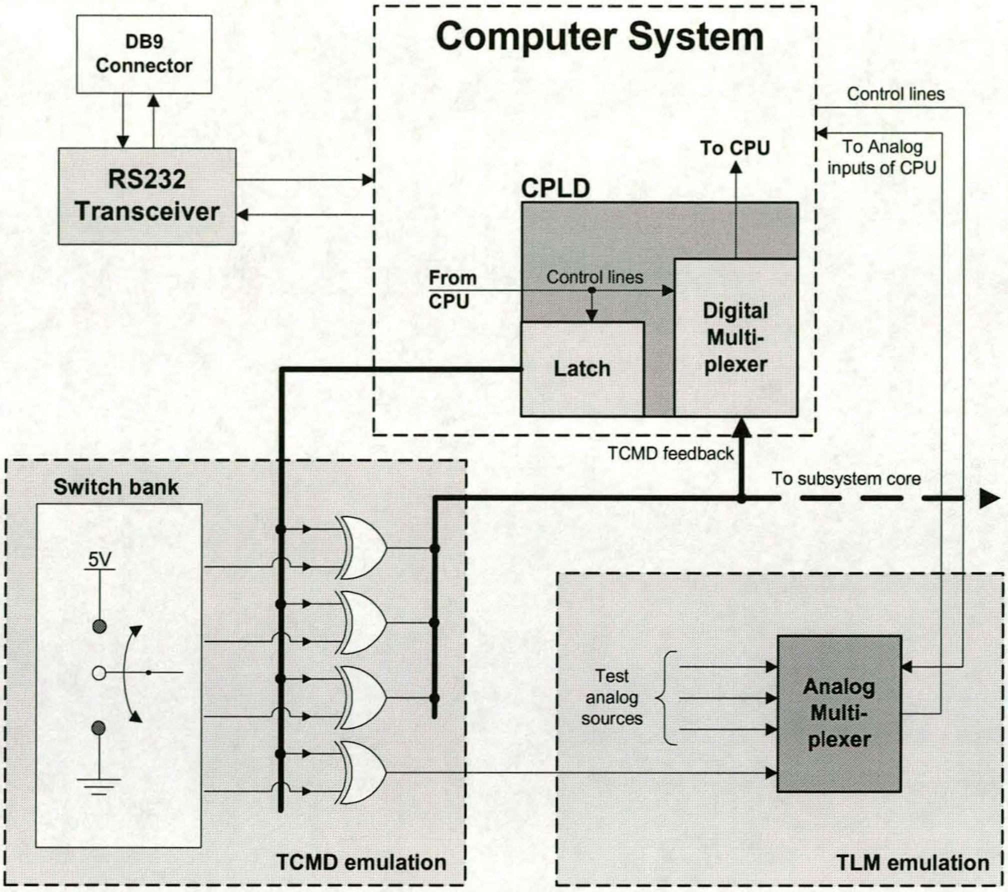

Figure 5.4 Data interfaces of the C&DH node

This is done before altering the input of the XOR gate connected to the C&DH node. When the target subsystem wants to set a particular TCMD, it can request the current state of the switch from the C&DH node via the communication processor, for instance. To emulate the setting of TCMDs on the subsystem core, four XOR gates are externally connected to the CPLD and a bank of switches. The outputs of the XOR gates are routed to the CPLD, where they are multiplexed and passed on to the CPU. Eventually they will also be connected to the subsystem where the particular node is situated. The CPLD also controls the inputs of the XOR gates under command from the CPU. At this stage, it is still uncertain precisely where the gates will be located. It may be either on the C&DH node itself, or on the particular subsystem. A separate IC is thus used here for the gates

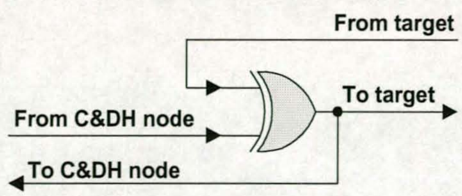

Figure 5.5 TCMD switch implementation

instead of implementing their functionality in the CPLD. The bank of switches is used to simulate inputs from the particular subsystem as shown in Figure 5.5.

TLM emulation

This section of the data interfaces is used to test TLM acquisition on the C&DH node. Three analog sources (multiturn potentiometers) are connected to an analog multiplexer (74HC4051), the output of which is connected to one of the eight analog inputs of the C515C, where it is sampled with 10-bit accuracy. The operation of the multiplexer is controlled directly from the CPU.

The output of one XOR gate is also treated as an analog source. This is implemented on the test setup in order to examine the possibility of sampling the status of TCMDs, instead of reading it in directly through one of the I/O ports of the CPU.

5.1.3 Communication interface

This part of the test C&DH node contains the CAN transceivers and related interface to the CPU. Figure 5.6 shows the components involved, the details thereof following below.

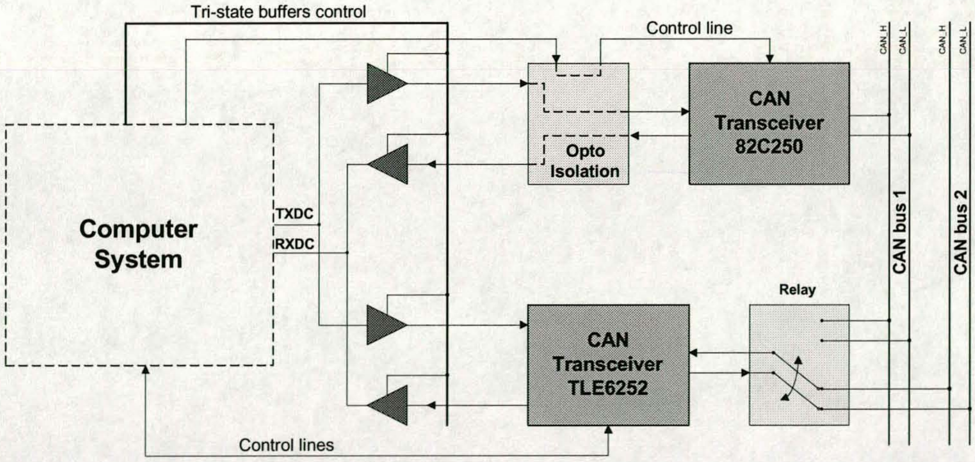

Figure 5.6 Communication interface of the C&DH node

CAN transceiver 1 - The Philips 82C250

This device provides differential transmit capability to CAN bus 1, and differential receive capability to the CAN controller on the C515C. It has the following features:

- High speed (up to 1Mbps);
- Low current standby mode (170µA maximum);
- At least 110 nodes can be connected to the CAN bus;
- Supply current (dominant bit) = 70mA maximum; and

- Supply current (recessive bit) = 18mA maximum.

Apart from the CAN-related functions, this 8-pin device (PCA82C250) has only one extra control line which is connected to the CPU, and is used to switch the driver into sleep mode.

CAN transceiver 2 - The Infinion TLE6252

This transceiver can be connected under command from the CPU, to any one of the two CAN buses via the relay, as shown. By default, it is connected to CAN bus 2. The TLE6252G is a fault-tolerant transceiver:

- Maximum transmission rate is 125Kbps.
- Switch is made to single-wire mode during bus line failure events.
- Prevention from bus occupation in case of CAN controller failure.
- Very low current consumption in stand-by (500µA max) and sleep mode (30µA max).
- Supply current (dominant bit) = 20mA maximum.
- Supply current (recessive bit) = 10mA maximum.

The 6252 has four additional pins that interface to the CPU: one is used to indicate a observed bus error condition; two to control the states of the transceiver (normal operation, go to sleep, standby and sleep) and a fourth line is used as a wake-up signal for the transceiver.

Opto isolation - The Hewlett Packard HCPL0600

Opto isolation is implemented between the Philips transceiver (82C250) and the CPU as is recommended by most fieldbus specifications. This is done in order to protect the digital control circuitry from power-system faults and ground loops. Furthermore, the reduction of common-mode noise and elimination of false signals ensures the integrity of data. The HCPL0600 has the following major features:

- +5V CMOS compatibility;
- 100ns max. propagation delay; and
- 7.5mA average typical current consumption.

The short propagation delay meets the transmission needs of the fast Philips transceiver. No isolation is provided for the TLE6252 transceiver. This is because the 6252 is a new device on the market, and is used here to evaluate its functionality. Double the number of opto isolators compared to the three used with the 82C250, would be required for complete isolation. Due to this, and the fact that the C&DH prototype is tested in a controlled and protected environment, no isolation is used for this transceiver on the test board.

The C515C has only one CAN controller on-board, and buffering is therefore needed between the controller and the two transceivers used on the test node. A 74HC125 tri-state CMOS buffer is used on each transmit and receive line. These four buffers are controlled by the CPU via the CPLD and are mutually exclusive: ie. only one transceiver will be connected to the CAN controller at any given time.

5.2 Memory mapping

The Infinion C515C memory map for the C&DH node can be divided into program memory and data memory. The latter can also function as program memory.

5.2.1 Program memory

The SAB-C515C-LM processor has no internal program memory, therefore the only available program memory on the prototype board is that provided by the flash and SRAM devices. During flash programming, program execution can be transferred from the flash to the SRAM. For development purposes, only address line 15 is decoded to transfer execution, as shown in Figure 5.9. The effective total program memory space available at any stage is therefore reduced to 32KB. However, an I/O line of the CPU is connected to pin A16 of the flash so that the upper portion of the device can also be accessed when needed. The program memory map is shown in Figure 5.7.

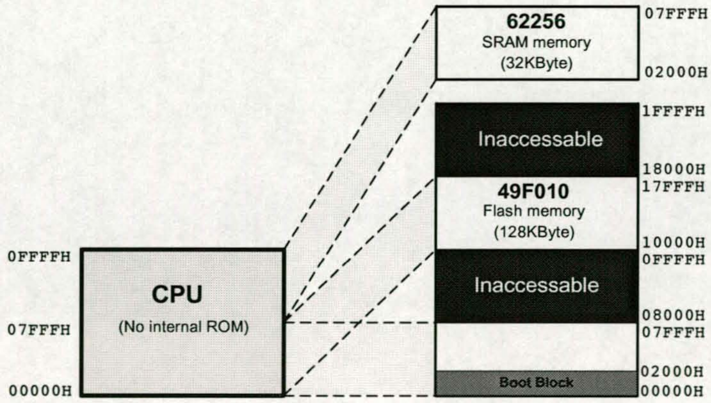

Figure 5.7 C&DH node program memory map

5.2.2 Data memory

A map of the data memory is shown in Figure 5.8. There are two external devices addressable in the data memory area: the flash and SRAM. Apart from the external data memory, the C515C also has two small separate blocks of internal RAM in the upper part of the 64KB memory space: 2KB of general purpose RAM, and the CAN interface with 256 bytes of RAM. These two blocks of internal memory overlap with external memory in the same address space, and therefore a special function register is allocated to assign one of the two blocks to the physical address space. The internal RAM is addressed with *MOVX* instructions in the same way as external RAM.

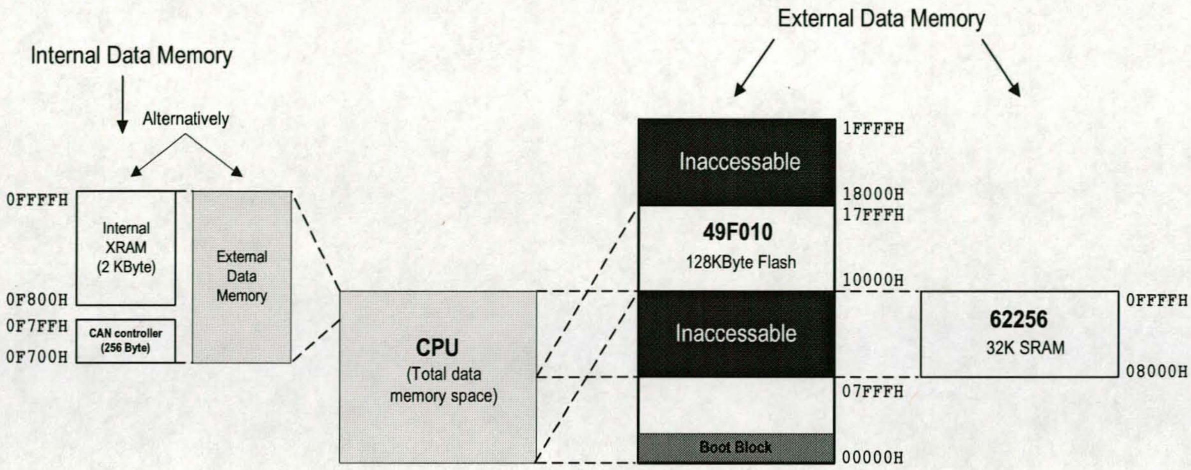

Figure 5.8 C&DH node data memory map

The detail of the address decoding, all of which is implemented in the CPLD, is shown in Figure 5.9. The flash memory will be enabled while the CPU is running ($\overline{\texttt{CPUR}}$ active), but the SRAM device must be enabled separately by asserting $\overline{\texttt{SEL\_RAM}}$. $\overline{\texttt{PSEN}}$ will automatically be gated to the SRAM when $\overline{\texttt{A15}}$ goes high. The same applies to the output enable and write enable signals for the SRAM.

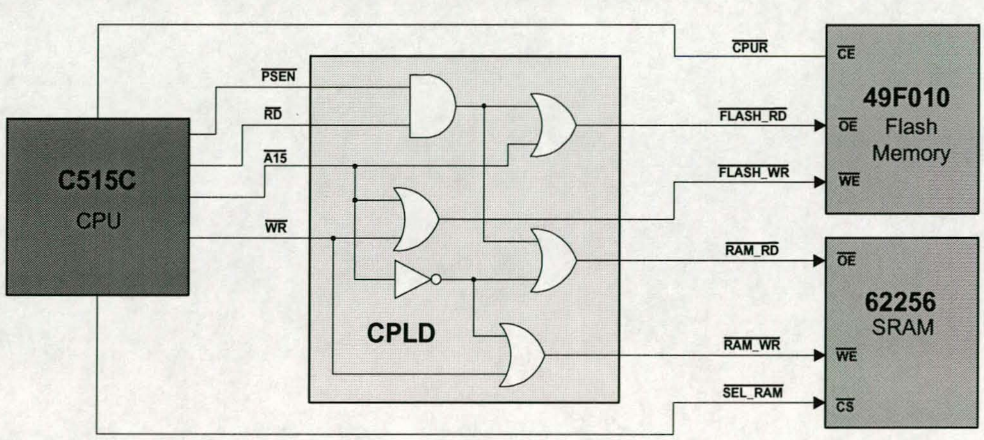

Figure 5.9 C&DH node address decoding circuitry

5.3 Timing analysis

The timing analysis of the C&DH node can be examined from the viewpoint of the C515C micro controller. Due to the lack of the appropriate *Synopsys* SmartModels™, a comprehensive timing analysis could not be undertaken for the C&DH node computer system. The only potential critical timing considerations on the board though, are with respect to the external memory components and the CAN transmission path. For the former, the applicable calculations can easily be done by hand. A worst-case assumption is made where the logical gates used in the CPLD design are replaced by discrete gates. An average maximum propagation delay of 25ns for the latter are then used in the calculations below. Calculations for the transmission path are discussed in Chapter 7.

The three main areas of interest in this section are the external program memory read, external data memory read and external data memory write cycles.

5.3.1 External program memory read cycle

Figure 5.10 shows the external program memory read cycle timing of the C515C processor. The important timing in this case is the output enable (OE) to output delay of the flash and SRAM devices. In Figure 5.9, it can be seen that the program store enable ($\overline{PSEN}$) signal from the CPU is delayed through two logic gates. The maximum total output delay is the sum of the propagation delay of $\overline{PSEN}$ and the output enable to output delay of the memory devices. The latter is the same for both devices:

$$t_{LOGIC}(\max) + t_{OE\,(flash,\,SRAM)}(\max) = 50ns + 40ns = 90ns \qquad \text{Equation 5.1}$$

This is less than the minimum width of the C515C $\overline{PSEN}$ pulse when running at maximum clock speed:

$$t_{PLPH}(\min) = 115\text{ns} \qquad \text{Equation 5.2}$$

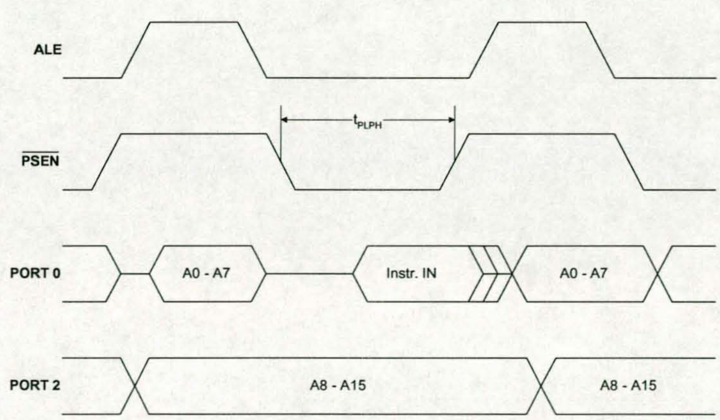

Figure 5.10 External program memory read timing diagram

5.3.2 External data memory read cycle

Figure 5.11 shows the external data memory read cycle timing. The two devices which can be accessed as data memory are the 49F010 flash and the 62256 SRAM. The important timing here again is the output enable to data output delays. The maximum output delays for these devices are the same, and Equation 5.1 therefore applies for this analysis as well. The output delays are both less than the minimum width of the C515C data read signal pulse and is also less than the time from $\overline{\mathrm{RD}}$ to valid data in (t_{RLDV}):

t_{RLRH} (min) = 230*ns* t_{RLDV} (max) = 150*ns* Equation 5.3

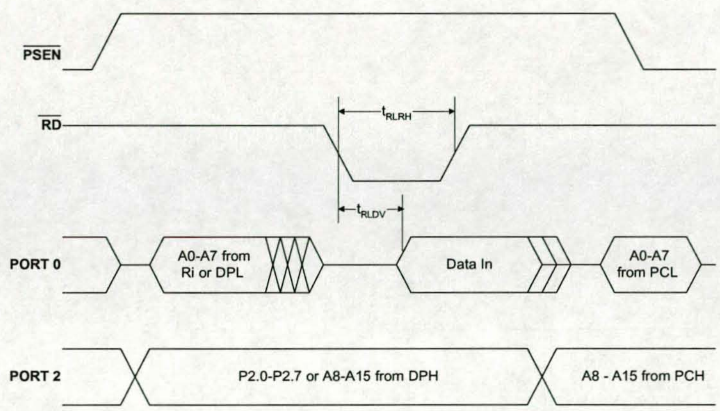

Figure 5.11 External memory read timing diagram

5.3.3 External data memory write cycle

Figure 5.12 shows the external data memory write cycle timing. The important times here are the minimum write pulse width, and minimum data set-up time requirements for the data memory devices. The latter requirements for the two memory devices are:

49F010: t_{DS} (min) = 50*ns* Equation 5.4

62256: t_{DW} (min) = 30ns Equation 5.5

The time that data is valid before the data write goes high, is the sum of the data valid to write low time and the write pulse width in Figure 5.12:

t_{QVWX} (min) + t_{WLWH} (min) = 5*ns* + 230*ns* Equation 5.6

This is higher than the required data set-up times. The other requirement is the minimum write pulse width of the C515C:

49F010: $t_{WP}(\text{min}) = 90ns$ Equation 5.7

62256: $t_{WP}(\text{min}) = 50ns$ Equation 5.8

These are less than the minimum data write pulse of the C515C:

$t_{WLWH}(\text{min}) = 230ns$ Equation 5.9

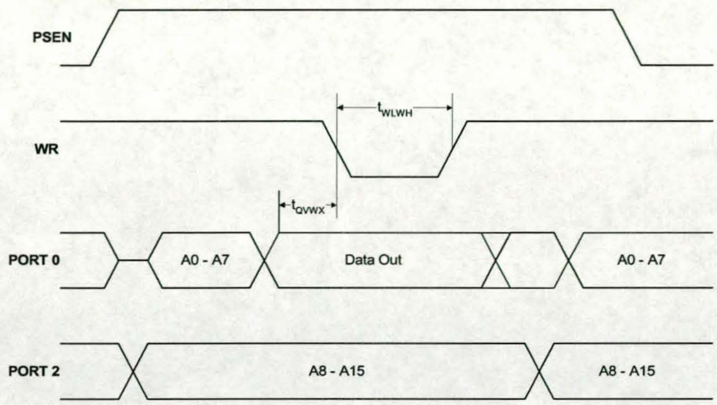

Figure 5.12 External data memory write timing diagram

As can be seen from above findings, all external memory devices are compatible with the timing requirements of the C515C running at 10MHz.

5.4 Conclusions

The two most important design requirements discussed before, namely a flexible TLM system and a TCMD system providing 100% feedback, have been met with the implementation of the test C&DH node presented here. Furthermore, the physical layout of the four-layer C&DH test PCB (see Appendix B) have dimensions of 15cm x 9cm. The final size of a C&DH node will be much smaller though, since the DIP packages used here to ease testing, can be replaced with surface mount equivalents. Also, a lot of functionality incorporated into the prototype board can be omitted on the final version where these test functions will not be needed.

Although the Infinion processor used here is very well suited for the application, an alternative may have to be investigated in future. This is necessary since component sourcing proved to be a big obstacle in the prototype development process. The SSC of the C515C device, though, is a good choice for an interface between the C&DH node and communication processor (see Figure 4.3). Its implementation falls outside the scope of this investigation and will also depend on the details of the communication processor. This has not yet been finalised.

TCMD lines can be expanded in future by using a larger PLD, such as an FPGA. Depending on the number of switches needed on each subsystem, the actual XOR gates could also be implemented in such a PLD, although the added component-level redundancy offered by external gates will also have to be taken into account. The power consumption of a programmable logic device will also have to be compared to the power consumption of an equivalent set of external gates. Expansion of the TLM acquisition capability of a C&DH node is facilitated by using external analog multiplexers on one or more of eight analog inputs of the C515C,

In future, a bigger SRAM device can be used to expand the WOD storage space of the node. This, however, would require a redesign of the full memory space. Furthermore, a separate voltage regulator is used to power the CAN transceivers on the board. The aim is to use this, together with full opto-isolation, in order to galvanically isolate the CAN bus from the digital side of the node. The ideal solution would then be to distribute power over two separate lines on the CAN bus, to all CAN nodes.

Chapter 6

International TLM and TCMD standards

A well-designed C&DH system on-board a spacecraft would be of little use unless it can also interact in an efficient and organised manner with one or more ground stations. This chapter therefore presents various international standards that could be implemented in future to ease ground-space interaction on SUNSAT-2, and also to raise the international esteem of the emerging South African space program.

6.1 CCSDS standards

The main international standards used on spacecraft today are formulated by the Consultative Committee for Space Data Systems (CCSDS), and are therefore the focus of attention in this chapter. Other international standards are mentioned, but are not discussed in detail.

6.1.1 CCSDS history

The CCSDS is a multi-space agency group, with 29 members and observers from North America, Europe, Japan and elsewhere around the world. The committee was established in the early 1980s to assist in standardising the space/ground links of the various agencies in order to increase the interoperability of their spacecraft and communication systems. The CCSDS has established recommendations for telemetry, telemetry coding, commanding, time codes, data formatting and radio frequency and modulation. These recommendations serve to guide the internal development of standards within the participating space agencies. More than 20 recommendations have been approved, and many have since also been adopted as ISO standards (see References for allocated ISO numbers).

6.1.2 Telemetry standards

The CCSDS TLM system is broken down into two major conceptual categories: a packet TLM concept and a TLM channel coding concept. Two recommendations have been developed for the former. The first, packet TLM [CCSDS, 1995], was developed in 1984. The second, Advanced Orbiting Systems (AOS), was developed in 1989 for potential application on missions such as the space station. This accommodates a more diverse set of data types, including voice and video.

The main aims of using CCSDS packet TLM are to [CCSDS, 1987a]:

- define a logical interface and protocol between an instrument and its associated ground support equipment;
- simplify overall system design by allowing microprocessor-based symmetric design of the subsystem control and data paths (TCMD packets in, TLM packets out)[1];
- eliminate the need for mission-dependant hardware and/or software with consequent cost and performance advantages; and
- facilitate the inter-operability of those spacecraft whose TLM interfaces conform to CCSDS guidelines.

The structure of CCSDS packet TLM is defined using the five lower layers of the ISO OSI layered model. Figure 6.1 shows how these layers - excluding the physical layer - map onto one another to encapsulate the raw TLM output from a CCSDS compliant spacecraft.

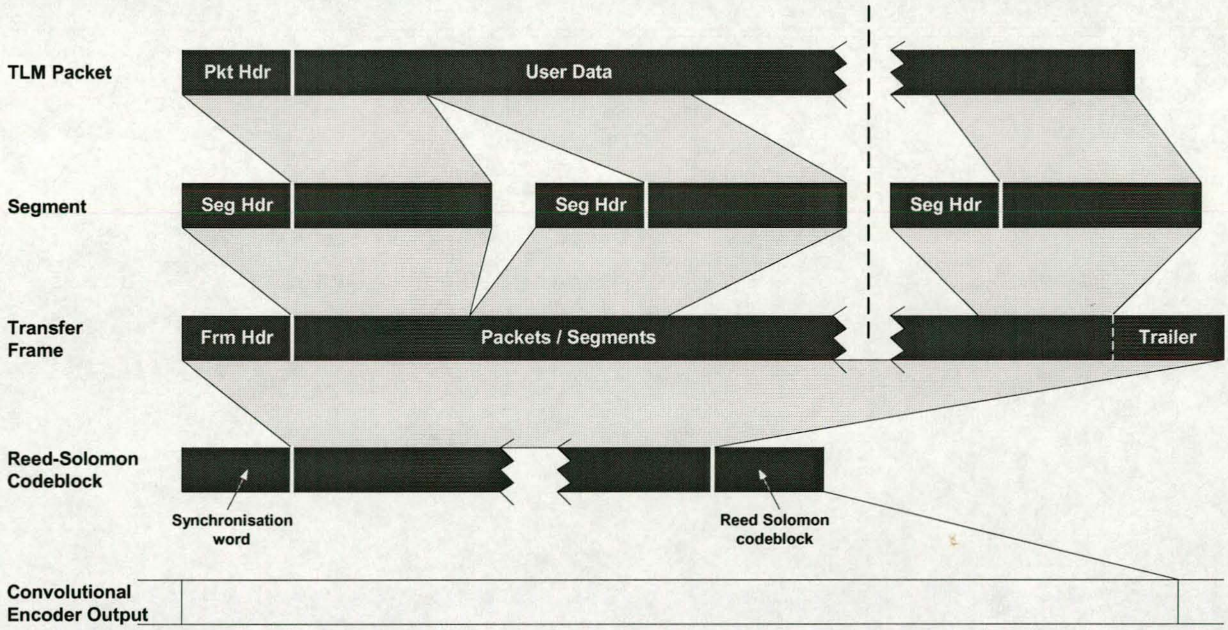

Figure 6.1 CCSDS telemetry data structures

Packetisation layer: A TLM source packet is the basic data unit telemetered to the user by the spacecraft. It generally contains one or more measurements of a particular application process. Appendix C contains details of the source packet.

---

[1] Such a design approach is compatible with commercially available components and interconnection protocol standards, like CAN for instance.

Segmentation layer: Large source packets can be segmented into communication oriented source packets called TLM segments. This flow-control mechanism can be used to prevent monopolisation of the data channel by very long source packets.

Transfer frame layer: The heart of the CCSDS TLM system is formed by the transfer frame which is used to reliably transport source packets and segments through the TLM channel to the receiving ground station. The transfer frame also accommodates the use of virtual channels on the same physical channel to differentiate data types, for example real-time data from recorded data (see Appendix C for further details). This is another flow-control mechanism.

Channel coding layer: TLM channel coding is used to protect the transfer frames against TLM channel noise-induced errors using one of the channel coding schemes in the TLM channel coding recommendation [CCSDS, 1992]. This is in line with a basic system requirement of error-free delivery of the TLM transfer frames. The output of the convolutional encoder - the synchronisation marker plus transfer frame plus Reed Solomon codeblock - is known as a channel access data unit (CADU).

Figure 6.2 shows the flow of TLM data from application sources on a spacecraft to corresponding destination sinks on the ground, implementing the layers discussed above. Note that an application process, corresponding to one application process identifier in a TLM source packet, can have more than one independent source of TLM data.

6.1.3 Telecommand standards

"The CCSDS telecommanding architecture defines a comprehensive set of layered, standardised command services which are applicable to a very wide range of mission needs" [CCSDS, 1987b]. As with the TLM recommendations, those of the TCMD are based on the layered network architecture, but were defined at a later stage and are also much more expansive. All seven network layers are covered, divided into three different documents[2] [CCSDS, 1987c,d,e] as shown in Figure 6.3.

At this stage, many of the functions within the data management service, with the exception of the packetisation layer, have been left for future specification. The aim of the total system is to provide the user with reliable and transparent delivery of TCMD

[2] A fourth document covers the topic of command operation procedures, and contains detailed information on retransmission protocols associated with the data routing service.

information, but since each layer is independent of the layers above it, individual missions may make their own decisions concerning how "high" in the layered hierarchy they wish to remain compatible.

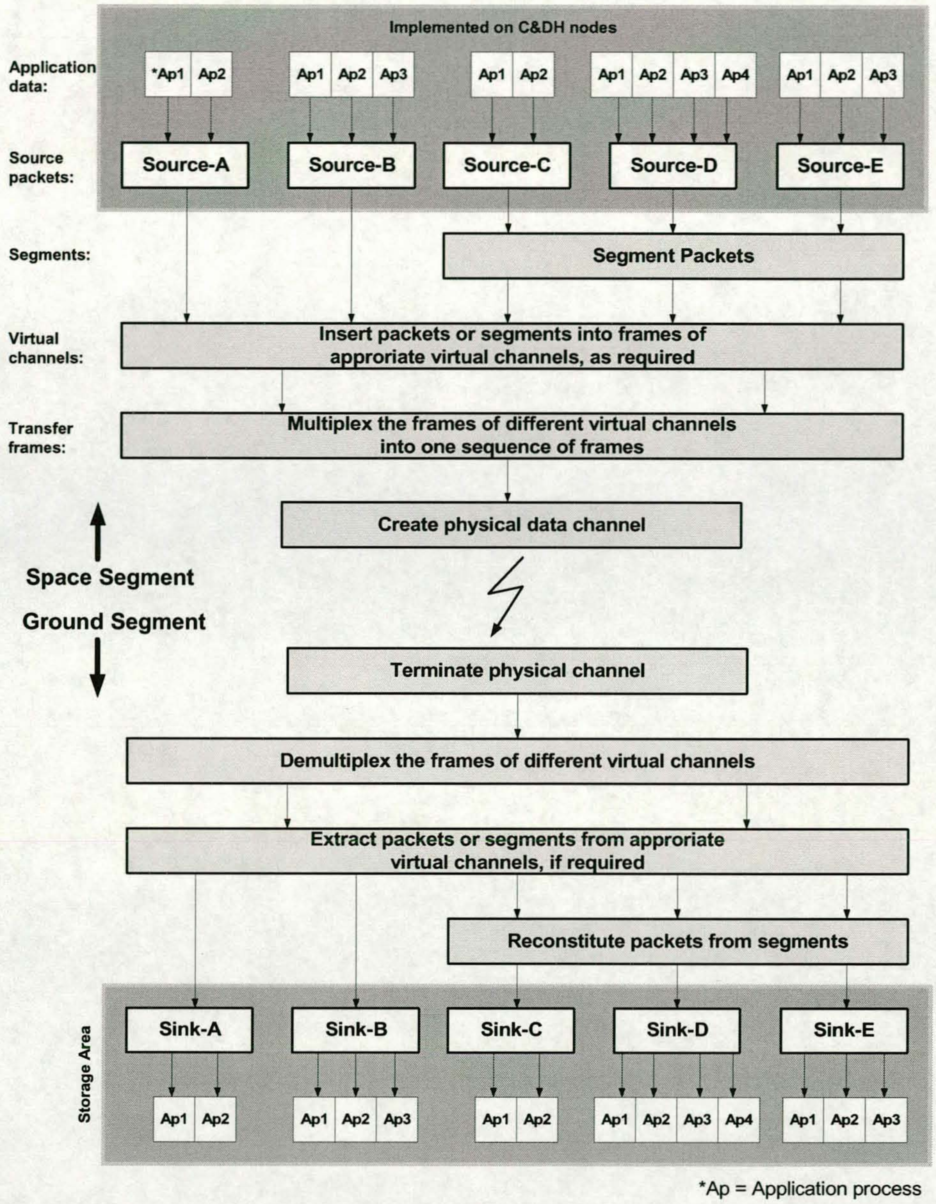

Figure 6.2 CCSDS telemetry data flow

The function of the TCMD layers are very similar to those of the TLM system discussed above. Their major functions are outlined below.

Application process layer: translates a user request into a high-level command which can be interpreted by the underlying system management layer. It therefore insulates the user from the physical aspects of the command delivery process.

System management layer: translates a high-level command into an interpretable TCMD data unit. It also provides control instructions to lower layers.

Packetisation layer: formats TCMD data into transportable data units called packets which are virtually the same as TLM packets.

Segmentation layer: breaks TCMD packets that are too long for effective handling by lower layers into smaller communication-oriented pieces. When the TCMD packets are short enough, the entire segmentation layer may be bypassed.

Transfer layer: uses its own data structure, the transfer frame, to transport TCMD packets or segments through the TCMD channel to the spacecraft. As with the TLM services, it can also format the packets or segments into appropriate virtual channels.

Coding layer: protects the TCMD user data against noise-induced errors during transmission through the ground-to-spacecraft RF channel. This is done by appending small blocks of information with parity bits that provide error detection and optionally

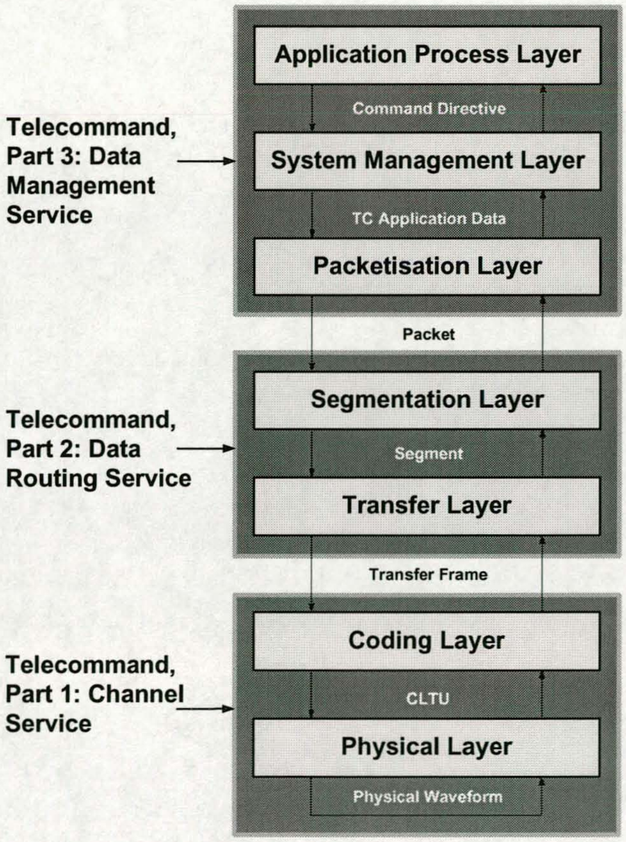

Figure 6.3 CCSDS telecommand document structure

correction capability. These code blocks are encapsulated into a command link transmission unit (CLTU) before being passed to the layer below.

Physical layer: modulates the CLTUs onto the RF channel.

As with the TLM standards, there are also two basic data structures used during the transport of a TCMD message from the ground to the spacecraft. The first is a TCMD packet and the second a TCMD transfer frame, the details of which can be found in Appendix C. Figure 6.4 shows how the structures mentioned above map onto one another. A command link control word (CLCW) can be used by the spacecraft to report command transfer status and verification information to the ground station via the TLM system. The CLCW is periodically sampled by the TLM and is returned via the CLCW slot in the trailer of the standard CCSDS TLM transfer frame (see [CCSDS, 1987c] for further details).

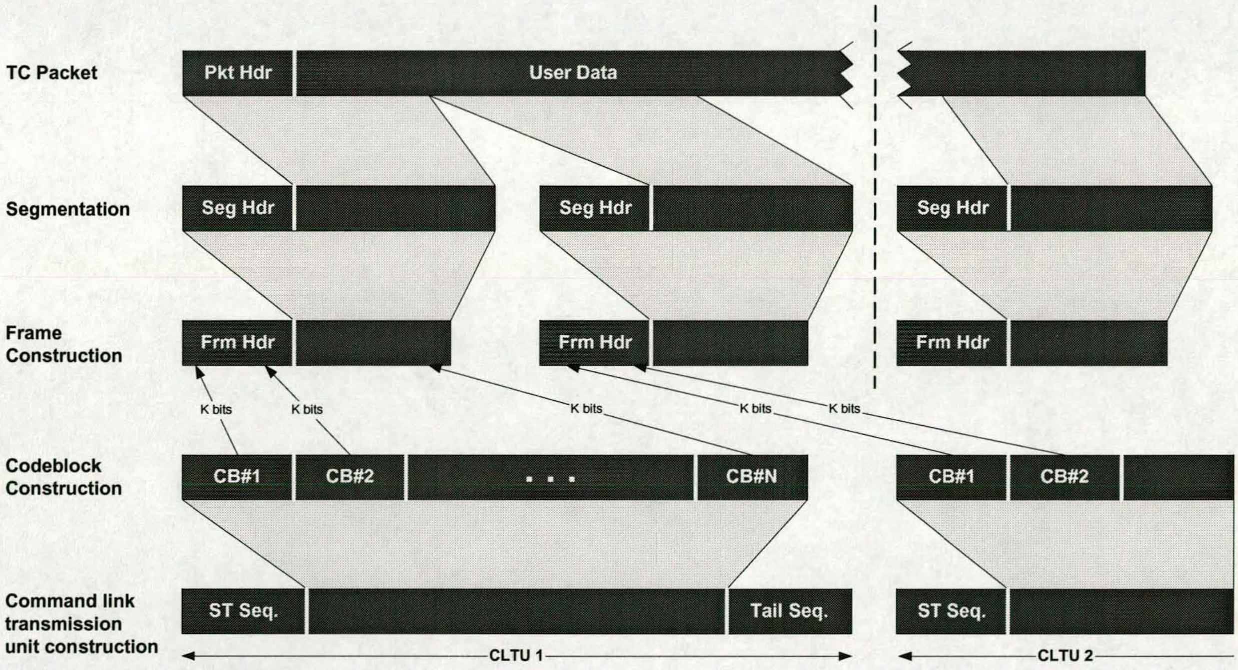

Figure 6.4 CCSDS telecommand data structures

The CCSDS also supports encrypted authentication[3] and data encryption[4], but does not have a corresponding recommendation. The choice of algorithms is left to the individual

[3] The ground station encapsulates each block of user data with a unique authentication word. The actual command data is left unchanged.

[4] Command data is transformed to make it unintelligible to an unauthorised observer, thus providing data protection.

space agencies, but the CCSDS does provide system-level requirements, for example to which level the algorithms should be constrained.

6.1.4 On implementing the CCSDS standards

Implementing the CCSDS standards on SUNSAT-2 would be a big advancement from the custom protocols used on SUNSAT-1, but not without some extended effort and the necessary precautions. The research presented here covers only the basics of these sophisticated protocols - a detailed study and implementation suite are left for future research.

Although the use of the CCSDS recommendations on spacecraft has been found to be "extremely positive" [Johns and Krimchansky, 1997] in many cases, some benefits along with pitfalls are listed below.

Advantages

- The system used for integration and test, and for operations, can be reused on other missions that use the same data system standards [Tompkins et al, 1997]. This implies large cost savings, and a reduction in operations risks since flight operators can re-apply the experience gained during previous missions.
- Using the CCSDS recommendations lowers the risk of the protocols being incomplete or having unforeseen consequences, because it has been evaluated in detail by experts from many space agencies, and has been used by other spacecraft.
- CCSDS formatted TLM allows the mission to be much more flexible. For example, on-board TLM is communicated among subsystems with only a subset of the packets provided to the TLM downlink. Note that the concept of CCSDS TLM packets and CAN frames containing TLM data starts to intertwine here. In paragraph 6.4 below a possible relationship between the two is presented.

Drawbacks

- The CCSDS recommendations are detailed but they are vulnerable to individual interpretation. Two different organisations can take the same recommendation and develop systems that will not interoperate completely. The recommendations should therefore be implemented in consultation with CCSDS experts [Tompkins, et al, 1997].
- A lesson learnt from the development of SUNSAT-1 is that the space segment was developed for the majority of the project, independently from the ground segment. In fact, parts of the ground segment were hastily put together a couple of months prior to launch, and could only be tested with the satellite already in space. To make most

effective use of CCSDS or any standard for that matter, any future mission has to be designed end-to-end, since the CCSDS recommendations also affect the ground segment.

- The recommendations include a variety of options and features to choose from. For a future SUNSAT mission, a subset of these has to be carefully selected and if interoperability with other organisations is required, the subset needs to include the features required to work with the other organisation's equipment.
- The overhead associated with CCSDS has to be taken into account when determining the link budget. For example, the average packet header overhead could be in the order of 15% for a typical application [Day, 1995], depending on the size of the data packets.

6.1.5 CCSDS examples

The number of spacecraft that have CCSDS assigned IDs amounts to over 170 (see Appendix D for a complete list of assigned IDs)[5]. It should also be noted that the CCSDS recommendations are not restricted to any spacecraft size or type of mission, although only intended for peaceful operations. They are currently used on microsatellites (Ørsted), large spacecraft (EDOS), earth orbiting, and deep space missions (Cassini).

The Jet Propulsion Laboratory (JPL) in Los Angeles, USA, has been developing a flight system testbed (FST) to serve as a spacecraft prototyping tool. It also acts as a reference flight system to implement and evaluate higher level spacecraft functionality. In this regard, they have developed a CCSDS protocol stack which they promote for re-use. This library of C routines (libCCSDS) could be used, in cooperation with JPL, to develop a test implementation of the CCSDS recommendations for SUNSAT-2. libCCSDS has been successfully applied on the Mars Pathfinder mission.

6.2 Other international standards

The CCSDS standards are the most widely used and accepted throughout the world. Others which are of particular importance are considered briefly below.

6.2.1 European Space Agency (ESA) standards

The ESA TLM and TCMD standards are intended to be used by the 15 European Union

[5] This number includes all past, present and some future spacecraft developments.

member states[6] and all other spacecraft utilising the ESA ground network facilities and services. These standards are directly derived from the corresponding CCSDS recommendations, and as such provide a high level of commonality between ESA equipment and that used by the other CCSDS-compliant agencies.

There are two **ESA TLM standards**: packet telemetry [ESA, 1988] and telemetry channel coding [ESA, 1989]. The packet TLM standard provides the same data structures as the corresponding CCSDS standard [CCSDS, 1995]. The only exception is the length of the secondary header data field in the TLM transfer frame (ESA secondary header = 4 bytes maximum; CCSDS = 64 bytes maximum). Minor differences do exist in the meaning of the subfields within the different data structures.

The **ESA packet telecommand standard** [ESA, 1992] is based on the four corresponding CCSDS documents [CCSDS, 1987b,c,d,e]. Again, the same data structures are used to transport TCMDs between a ground station and a spacecraft. The only difference here is the total length of the transfer frame data field (ESA = 249 bytes maximum and CCSDS 1019 bytes maximum). The TCMD decoder specification [ESA, 1993] further specifies the interfaces and data formats for a packet TCMD decoder, including an authentication unit. Establishment of a similar document is planned for packet TLM encoders.

6.2.2 Inter Range Instrumentation Group (IRIG) standards

The applicability of the IRIG standards outside of the United States Department of Defence (DoD) is limited but are mentioned here since they are sometimes referred to in the literature [Larson and Wertz, 1992:382]. A Range Commanders Council (RCC) was founded within the DoD to develop operational procedures and standards for range use, including the space wings. The RCC has several standing technical groups, including a TLM group responsible for maintaining IRIG Standard 106 [IRIG, 1999]. This standard is not as comprehensive and widely used as the CCSDS and ESA ones, and will therefore not be discussed in any further detail.

6.3 Amateur standards for TLM downlink

The fact that SUNSAT-1 is an amateur satellite (designated as OSCAR-35, or SO-35 in short) includes the possibility that SUNSAT-2 might also carry amateur radio payloads.

[6] Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxemburg Netherlands, Portugal, Spain, Sweden and the United Kingdom.

For this reason, provision has to be made for suitable protocols to serve the needs of the amateur radio community. But as Kasser [1992] points out, the majority of OSCARs launched since 1961 used different data downlink formats, making it very difficult for the users to capture and process the data from these satellites. Even after eight months in orbit, no software has been made available to the general public to decode and display the 1200-Baud FSK TLM from SO-35. Finding a 'suitable protocol' can therefore be more involved than anticipated.

Approximately fifteen amateur satellites were operational in space [Ford, 1999] at the time of writing this document. Many of them carry 1200 and 9600 packet services, including SO-35. All the packet service satellites use the Amateur version of the X.25 protocol, AX.25, which is considered the de facto transfer protocol for amateur radio use [Jones, 1995]. But, the format of the TLM data encapsulated in an AX.25 packet varies widely. This is particularly evident when one looks at the suite of different SW available to decode and display TLM data from the amateur satellites[7].

The reason for the lack of a standard TLM protocol for amateur use is not apparent, but may be attributed to the small number of amateur satellites in use in comparison with the magnitude of commercial and scientific satellites. It would be wise to include at least the AX.25 protocol stack on SUNSAT-2 for possible radio amateur use, but also for general packet transfers. AX.25 is a well-established protocol with error detection capabilities and an extended performance record on SUNSAT-1.

6.4 C&DH nodes and the CCSDS protocols

The implementation of the international standards presented above has architectural implications on the proposed C&DH nodes discussed here. In Figure 4.1, a distributed architecture is shown for the on-board C&DH system, but the question of a suitable link between the ground and space segments to complete the overall C&DH chain still remains.

It would be impractical to implement the CCSDS standards locally on the satellite, due to the relatively large protocol overhead of CCSDS frames (see paragraph 6.1.4 above) and the 8-byte data length limit of CAN. Figure 6.5 shows a possible solution to this problem. A processing element is implemented on the communication subsystem (VHF for instance)

---

[7] For instance *What's Up* (for UO-9, UO-11, FO-12, AO-13, AO-16, DO-17, WO-18, LO-19, and FO-20) and *TLMDC4* (for AO-16, WO-18, and LO-19)

which formats all TLM received via the CAN local area network into CCSDS frames. It will also decode and authenticate CCSDS TCMD frames received from the ground and pass it on to the C&DH node for distribution on the CAN network. The C&DH distribution remains decentralised but the protocol implementation and ground link is centralised.

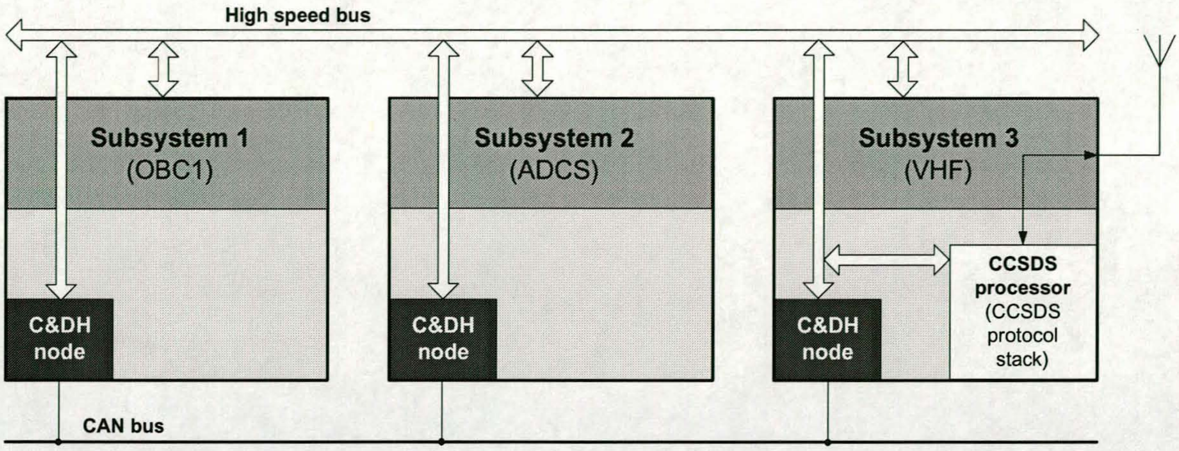

Figure 6.5 Relationship between C&DH nodes and a CCSDS processor

This C&DH processor can also filter incoming real-time TLM packets and interleave them with playback data received via the high speed bus for example, to form virtual channels. Such a filtering mechanism would be based on the application process identifier (APID) contained within the CCSDS TLM source packet header (see Appendix C). Note that it would be convenient to use the 11-bit identifier contained within the arbitration field of a CAN frame to form the 11-bit APID. However, this would only be possible if every CCSDS source packet is associated with only one application process, as opposed to the setup shown in Figure 6.2. Otherwise, if a one to one mapping of CAN IDs and application processes on a C&DH node is not practical, the eight data bytes within a CAN frame should be subdivided into a control portion and a data portion. The control portion can then be used to distinguish between different application processes on a C&DH node using the same CAN identifier.

Figure 6.6 shows the detail of a possible CCSDS processing implementation. Depending on the actual communication systems to be used on SUNSAT-2, the CCSDS processing could be duplicated on more than one subsystem (VHF, UHF and L-band for instance). A packet TCMD decoder[8] (PTCD) has been developed and used on various ESA spacecraft [Sinander and Habinc, 1994]. This radiation hard, latch-up free component accepts CLTUs from a RF receiver, passes it through an on-chip authentication unit, and

[8] Mitel MS13545

strips the CCSDS protocol overhead before passing it on for distribution. It also generates several status reports, including a CLCW for direct feedback to the ground station via the TLM encoder.

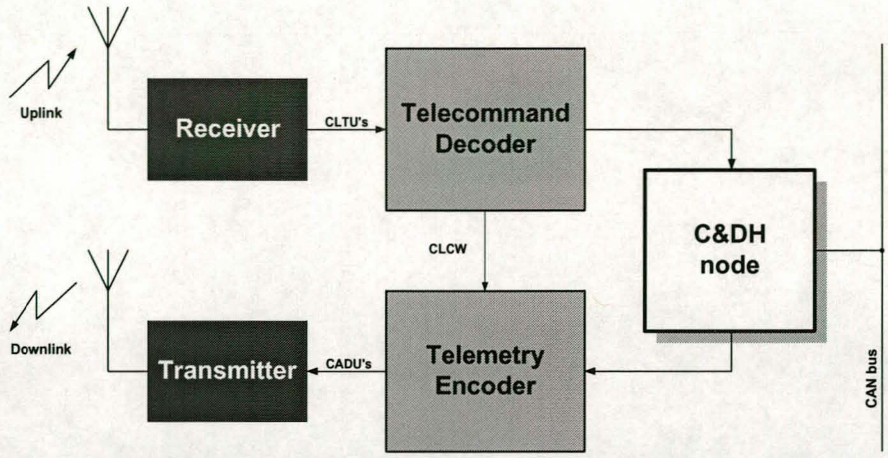

Figure 6.6 CCSDS processing implementation detail

The TLM encoder receives TLM packets from an on-board data source, and formats them into CCSDS frames. Optionally, it can also add Reed-Solomon and/or convolutional encoding before outputting a complete CADU to a RF transmitter. A typical TLM encoder consists of up to eight virtual channel assemblers[9], and a virtual channel multiplexer[10]. When availability[11] and cost of the above components are a problem, the TCMD decoder and TLM encoder may be implemented using a custom processor design.

6.5 Conclusions

This chapter looked at three international TLM and TCMD standards: those of the CCSDS, ESA and IRIG. The CCSDS standards are the most widely used throughout the world. The ESA ones, derived from the CCSDS standards, are mainly used by the European Union member states. IRIG TLM standards were briefly mentioned but their significance outside the DoD is very small. It is proposed that the AX.25 protocol be included on future satellites, not only for radio amateur use, but also as a reliable general packet transfer protocol.

[9] Mitel MS12399

[10] Mitel MS12396

[11] The cited Mitel parts have since been withdrawn from the market, but Saab Ericsson Space is developing replacement parts with possible higher level functionality.

Due to the short message length of a CAN frame, the chosen CCSDS protocol should not be implemented between the various subsystems on the satellite, but rather only between the satellite and ground station link. It is expected that this will increase the inter-operability of future SUNSAT satellites and therefore raise the international esteem of the project.

Part C
Evaluation

7. Parameter assessment
8. Future research
9. Conclusions

Chapter 7

Parameter assessment

As stated at the beginning of this document, the aim was not to design a complete and fully functional C&DH system. Only key aspects of such a system were implemented on the prototype design. In this chapter, these aspects and other performance parameters are evaluated against the set goals and specifications presented earlier.

7.1 Prototype performance

This section contains the most important physical measurements carried out on the prototype system. The measurements are listed here in order to highlight the importance of characteristics such as low power consumption, fault tolerant capabilities, and timing constraints.

7.1.1 Current consumption

Table 7.1 shows the current consumption of the most important components on the prototype board. Under normal operating conditions [1], the default current consumption was measured to be 146.5mA. The biggest contributors to this relatively high figure are the CPLD, opto-couplers, and the RS-232 transceiver.

A SW power down mode was invoked by an external event trigger (interrupt); switching the CPU, flash, SRAM and two CAN drivers into a sleep/power down mode [2]. This mode cut the total current consumption by 53.5mA. Also note that the CPU consumes 14.9mA less current when running at a clock speed of 2MHz instead of 10MHz. Furthermore, measurements have shown that the fault tolerant transceiver (TLE6252) consumes approximately one fifth of the current of the 82C250 transceiver when the CAN bus is in the idle or recessive state.

---

[1] CPU executing code at 10MHz; no CAN transmissions; no RS232 transmissions; SRAM disabled; no LEDs lit except power LED.

[2] It is possible to exit this mode by applying an active low signal at the $\overline{\text{INT0}}$ pin. On a running C&DH system, this can be accomplished by connecting the RXD output of for instance the fault tolerant CAN transceiver to the $\overline{\text{INT0}}$ pin. In sleep mode, the 6256 transceiver signals the CPU when a CAN frame has arrived, by taking the RXD output pin low.

Table 7.1 Prototype board current consumption

	Normal[7]			Low power[7,11]		
Component	**Measured**	**Typical[1]**	**Maximum[1]**	**Measured**	**Typical[1]**	**Maximum[1]**
C515C[8]	6.4	-	-	-	-	0.05
C515C[9]	21.3*	-	-	-	-	0.05*
49F010	18.2*	-	30	-	-	0.1*
62256	11*	33	60	-	0.0002	0.05*
Max232	12.94*	15	-	-	-	12.94*
EPM7064	51*	-	-	-	-	51*
82C250 R[2]:	12.5	-	18	-	-	0.17*
D[3]:	19.7	-	70	-	-	0.17
6262 R[2]:	2.7*	3.5	10	-	0.015	0.03*
D[3]:	- [4]	6	20	-	0.015	0.03
HCPL-0600	7.5*3*	6.5	-	-	-	7.5*3*
Relay	33	-	-	-	-	-
Other[10]	6.86*	-	-	-	-	6.86*
Total:	146.5[5]	-	-	93.0[6]	-	93.7[5]

Notes:
1 From the corresponding data sheets
2 Recessive
3 Dominant
4 This could not be measured, as the 6265 has a shut-down feature when the TXD pin is stuck at a logical low level.
5 All values indicated with a * are totalled.
6 Only the total board current was measured with all the applicable devices in a power down mode.
7 All values in mA.
8 CPU running at 2MHz clock speed.
9 CPU running at 10MHz clock speed.
10 Voltage regulator, power LED, logic components and pull-up/down resistors.
11 All applicable devices in sleep mode.

7.1.2 Fault tolerant transceiver characteristics

As stated before, the Infinion TLE6252 is a new fault tolerant CAN transceiver available on the market. It has a maximum transmission speed capacity of 125kbps. This limiting factor however, does not preclude its possible inclusion in future C&DH designs. Three main features that have been verified on the prototype board, are therefore highlighted below:

Transceiver shutdown

When the TXD pin of the transceiver is grounded, the device outputs a 194μs active low pulse on the NERR pin after the disable time period (t_{TxD}). This indicates that the transmitting stage of the device has been deactivated, and can no longer block the bus. Figure 7.1 indicates the measured parameters with typical values from the data sheet in brackets. To get a feel for the relative length of t_{TxD}, approximately 2½ standard 8-byte

CAN messages at 125Kbps can be transmitted in the same period.

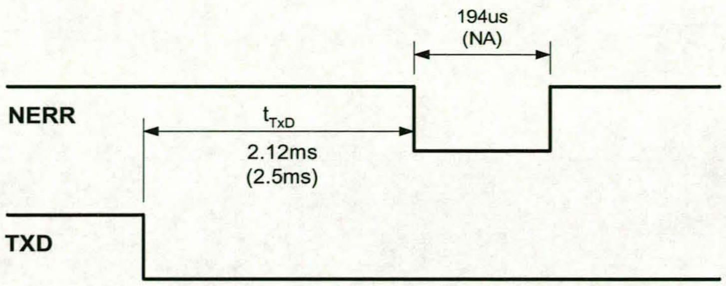

Figure 7.1 TLE6252 permanent dominant disable time

Bus failure management

The TLE6252 can detect bus failures and automatically switch to a dedicated CANH or CANL single wire mode to maintain data transmission. The failure conditions measured are shown in Table 7.2 below.

Table 7.2 Measured CAN bus failure states with the TLE6252

CAN_H	CAN_L	Current	Error counters increment?
Vcc	-	↓ ~2mA	Yes
Gnd	-	↑ ~30mA	No
-	Vcc	↑ ~30mA	No
-	Gnd	↓ ~2mA	Yes

The last column in the table shows the behaviour of the error counters during bus failure modes. The fact that the error counters will increment in certain cases is confirmed below in Figure 7.3. When CAN_H is shorted to Vcc during the transmission of a message, the remainder of the message will be discarded, causing the error counters to increment. This also happens when CAN_L is shorted to GND, but not for the other cases (see Figure 7.2).

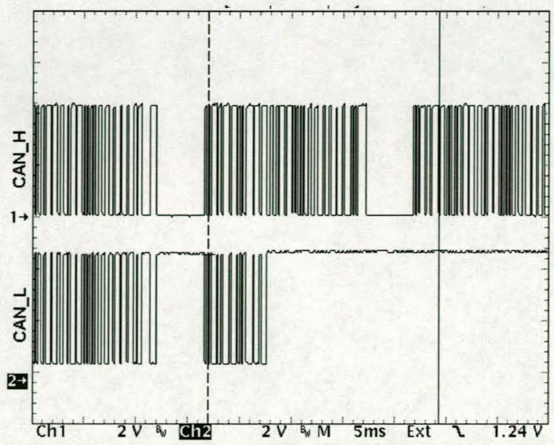

Figure 7.2 CAN_L shorted to Vcc

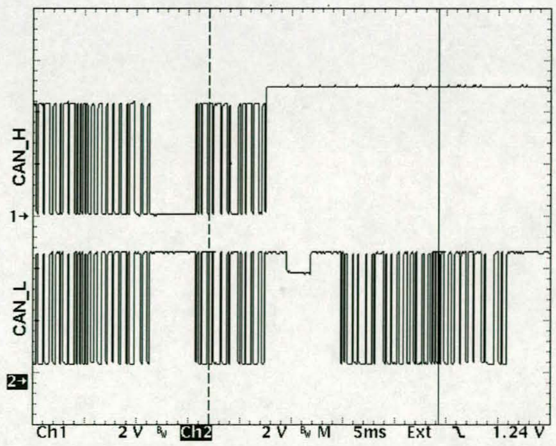

Figure 7.3 CAN_H shorted to Vcc

To eliminate false triggering of the bus failure modes, the CAN bus will only be switched over to a single-line mode after a certain delay time; and will only be switched back to differential mode after a delay when the bus failure condition disappears. Two other bus failure modes not shown in the table, are the interruption of either the CAN_H or CAN_L line. This could not be verified, as only one TLE6252 was available in the test environment. Bus failure modes cannot be simulated when the TLE6252 is connected to a non-fault tolerant transceiver.

Sleep modes

The TLE6252 has various low-power modes which can be controlled by two input pins. Upon a wake-up request either from bus line activities or the WAKE input pin, the device signals the CPU by taking the RXD pin low. The CPU must react to this condition by putting the device into normal operating mode. The time between a wake-up request and RXD=0 was measured to be 26μs, compared to the data sheet typical value of 22μs. At a bus speed of 125Kbps, the first message received will therefore be lost, as can be seen in Figure 7.4.

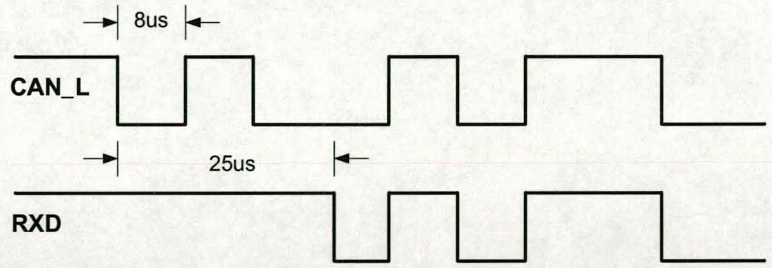

Figure 7.4 TLE6252 wake-up via CAN bus (bus speed = 125Kbps)

7.1.3 Timing considerations

In this section, three important timing parameters are evaluated. Firstly, the A/D conversion time is measured as this influences the maximum sample speed of TLM channels. Secondly, the maximum expected bus node loop delay is calculated, taking into account the propagation delay of all in-line components. The loop delay places a restriction on the maximum bus speed. Lastly, the latency of a typical message is calculated and compared to a physical measurement. This parameter is important, as the need may arise to set pseudo-real-time TCMDs with a very short delay time via the CAN bus.

A/D conversion time

Two independent measurements were carried out to verify the conversion time of the

C515C's 10-bit built-in A/D convertor. In the first measurement, an internal timer was started immediately before the start of a conversion cycle. The timer was then stopped by the first instruction of the A/D interrupt routine, which is called at the end of a conversion cycle. The resultant timer value indicated a conversion cycle time of 12µs, taking into account the fact that the first instruction of the interrupt routine is only executed at the fourth instruction cycle after the end of conversion.

In the second measurement, the BUSY flag was polled. This flag is set at the start of a conversion cycle and reset at the end of the cycle. Again, a timer was used to measure the cycle time, which was found to be the same as before. However, the measured value is bigger than the expected maximum conversion time given in the data sheet, namely 9.6µs. The absolute maximum theoretical TLM sample tempo would therefore be 50 kilosamples per second, if one estimates the conversion cycle time conservatively at 20µs.

Bus node loop delay

The bus node loop delay is defined as the time taken for a signal to travel from a CAN controller on one node, through all external components, over a physical bus to another node on the same network, and back again. For the following calculation, it is assumed that the maximum cable length on a micro satellite with SUNSAT-1 dimensions will not exceed 5m, excluding stub lengths. The propagation delays of the in-line transmission path components on the prototype board are listed in Table 7.3.

Table 7.3 Propagation delay of critical components on the C&DH prototype.

Component	Part number	Propagation delay (ns)
Three-state buffer	74HC126	16.5 x 2[3]
Optocoupler	HCPL-0600	100 x 2[3]
CAN transceiver	82C250	50 x 2[3]
Cable[4]	-	25
	Total:	358

The bus node loop delay is therefore 2 x 358ns = 716ns which translates to a maximum rate of 1.4Mbps on the bus. This rate is within the CAN specification of 1Mbps, but the

[3] These terms are multiplied by two in order to make provision for the component delays on the receiving node.

[4] For this calculation, it is assumed that the travel speed of an electrical signal on a copper wire is ⅔ the speed of light, or about 5ns/m.

safety margin can be increased further by using faster optocouplers, for instance.

Telecommand latency

For time-critical TCMDs, it is necessary to know what the latency[5], R_m, of a CAN message on the bus would be under nominal conditions. R_m is given by [Tindell and Burns, 1994]:

$$R_m = J_m + w_m + C_m \qquad \text{Equation 7.1}$$

where J_m is defined as queuing jitter of a given message *m*, dependant on the response time of the CPU. The term w_m represents the worst-case queuing delay of a message and C_m is the longest time taken to physically send message *m* on the bus. Ideally, Equation 7.1 can be simplified to:

$$R_m = C_m \qquad \text{Equation 7.2}$$

when it is assumed that there are no other messages being transmitted on the bus, or there are messages scheduled on the host CPU waiting to be transmitted. These assumptions are made in order to obtain a quick approximation of the expected latency under ideal conditions. For a standard CAN frame, Equation 7.2 then becomes:

$$R_m = \left(\left\lfloor \frac{34 + 8s_m}{4} \right\rfloor + 47 + 8s_m \right) t_{bit} \qquad \text{Equation 7.3}$$

where the term in square brackets is the theoretical maximum number of stuff bits; the second term is the protocol overhead, and the last term is the number of data bits being transmitted. The term t_{bit} is the bit time on the bus. Substituting values for a one-byte CAN message being transmitted on a bus running at 125Kbps, Equation 7.4 resolves to:

$$\begin{aligned} R_m &= (10.5 + 47 + 8) * 8\mu s \\ &= 524\mu s^{6} \qquad \text{Equation 7.4} \end{aligned}$$

During a practical measurement, two nodes were set up on a bus running at 125kbps. One node transmitted a message to set a TCMD switch on the receiving node. Under similar assumptions as above, the 'latency' was found to be 484µs. This value also includes the total propagation delay as listed in Table 7.3 as well as the time taken by the receiving node SW to set the actual TCMD switch. Still, it is less than the theoretical value

5 Latency is defined as the time taken from a transmission request for a particular message to the last bit of that message transmitted by the CAN controller.

6 When the bus is running at 1Mbps, this value becomes 65.5µs.

of Equation 7.4, due to fewer stuff bits in the practical case. Although the practical value cannot be compared directly to the theoretical value since the former includes additional parameters not accounted for in the definition of $R_{m,}$ they are of the same order.

7.2 Flexibility

As stated earlier, flexibility is an important requirement for the new C&DH system. This section evaluates this requirement against three key aspects implemented on the prototype board.

7.2.1 Code upload functionality

Figure 7.5 shows how the two memory components on the prototype board are mapped onto the program memory space of the CPU. The address decoding is such that code at addresses below 32KB (07FFFH) will be executed from flash, and those above 32KB from SRAM. The usable address space is therefore cut in half, but a different allocation between SRAM and flash memory space can be accomplished by decoding more address lines. The design shown in Figure 7.5 is necessitated by the fact that code cannot be executed by this particular flash device when it is programmed. A small portion of the code residing in the flash boot block is therefore copied over to the lower part of the SRAM memory. A subsequent jump instruction to address 08000H will transfer program execution to the SRAM. New program code can then be uploaded to the remainder of the SRAM before programming it into the flash. After successful programming, execution is transferred back to the flash device by a jump instruction to address 0H.

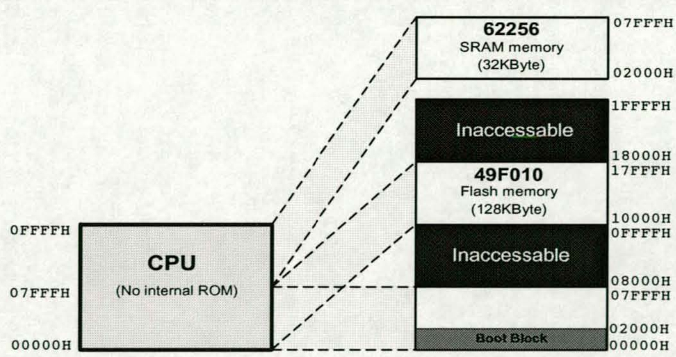

Figure 7.5 Program memory space of the C&DH prototype

The new program code can be uploaded to the SRAM via the CAN bus or via the communications processor module. Using Equation 7.2, and assuming an unloaded CAN bus running at 1Mbps, it is possible to calculate the approximate time it would take to upload a 32KB section of new code via the CAN network. It is further assumed that six bytes in each CAN frame are used to transfer actual code information and the remaining two bytes to control the upload process. Then,

$$t_{Upload} = R_m * \frac{32*1024}{6}$$
$$= 135.5\mu s * 5461 \qquad \text{Equation 7.5}$$
$$= 0.74s$$

7.2.2 Dual CAN network

The flexible architecture shown in Figure 7.6 allows a mixture of high-speed and low-speed communication within the same C&DH network architecture. For instance, when it is necessary to transmit TLM samples at high speed on the CAN bus, or when the busload becomes too high on the low-speed CAN bus, the CPU can direct communication to the high-speed bus. With such an arrangement, it is essential that TCMDs be broadcast on both CAN networks to ensure that all nodes receive TCMDs at any stage. A further advantage of the implementation shown in the figure below, is that the fault tolerant transceiver can communicate on both buses in the event of permanent or non-permanent failures on any of the buses.

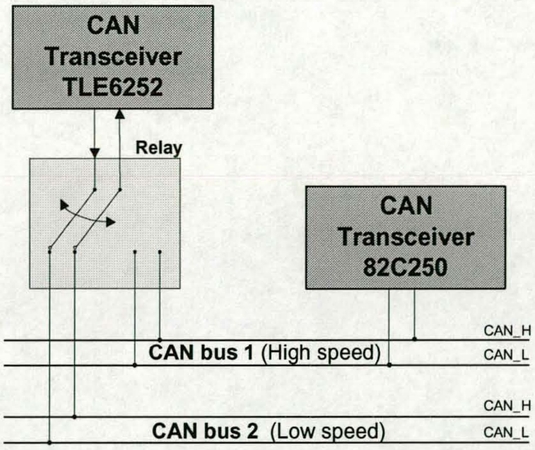

Figure 7.6 C&DH prototype dual CAN network

The relay used on the prototype board is an electro-mechanical device, but should be replaced by a magnetic latching relay[7] on a flight version to limit current consumption.

7.2.3 Mode changes

With the prototype implementation, it is possible to change the TLM sample speed per individual node as the need arises. Critical parameters or science experiments can then be assigned a higher sampling rate, while maintaining a low sampling rate for

7 For instance a Teledyne 420

housekeeping data or other slow-changing parameters. WOD can also be gathered and stored on local SRAM on each node, and can then be transmitted in high-speed bursts at the request of the ground station or OBCs. This reduces the busload, and therefore has a power saving advantage. It is also possible to define an 'idle mode' when only key housekeeping parameters need to be broadcast on the CAN bus at minute intervals, for instance. This will be beneficial during remote parts of orbit.

A further consequence of the flexible design is the degree of autonomy realised on each node. Local parameters such as temperatures and currents can be monitored continuously by each node without necessarily broadcasting the values on the bus. When such values exceed a defined limit, the node can take appropriate action by immediately issuing a TCMD for instance. Moreover, the fact that nodes on a shared bus are used to implement a C&DH system, makes it inherently flexible: new nodes can be added without having to adapt the system architecture.

7.3 CAN ID assignment

A very important aspect of CAN network management is the assignment of message priorities, or IDs. This directly influences bus arbitration, and also has a visible effect on bus loading, schedubility of messages, and message latency. An ID assignment scheme is proposed here to manage the flow of TLM and TCMD packets on a single CAN bus.

7.3.1 Proposed scheme

Using Equation 7.2, it can be shown that the worst-case maximum data content of a standard CAN frame is only 47.4%. For extended CAN frames having 29-bit identifiers, this figure drops to 40%. The data efficiency on a CAN network can therefore be maximised by using standard CAN frames and making full use of the allowable eight bytes of data.

Referring to the 11-bit arbitration field in Figure 7.7, the following is proposed:

- The most significant ID bit indicates whether the frame contains TCMD (T&T sel = 0) or TLM (T&T sel = 1) data. TCMDs therefore have the highest priority, since the first bit of the arbitration field is dominant.
- The next four bits are the node address making provision for 16 nodes.
- The next bit is reserved for future use and must be set to '0'.
- The TLM channel selector consists of four bits to select 16 different TLM channels. If the frame contains TCMD data, this field has no particular meaning and must be set to

'0101' to minimise stuff bits.

- The last bit is also reserved and must be set to '0'.

The above is a form of deadline monotonic priority selection where it is assumed that TCMDs will always have the shortest deadlines and will therefore be assigned the highest priority.

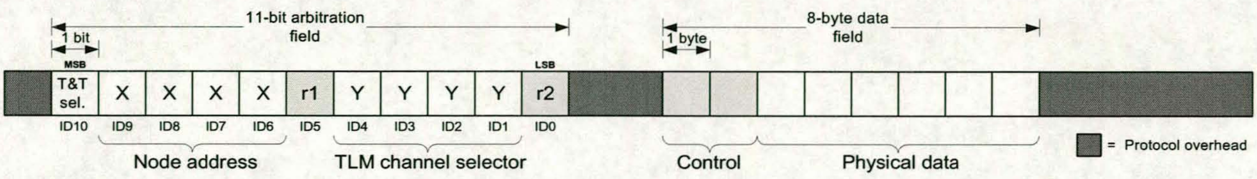

Figure 7.7 Break-up of a standard CAN frame on the proposed C&DH system

The data field of each frame is subdivided into a two-byte control portion and a six-byte data space. The control portion is included to make provision for a subprotocol within the CAN network, and remains undefined at this stage. In the case of a TLM frame, six bytes of data can represent four 10-bit TLM samples if a piggybacking technique is employed. The TLM channel selector then indicates the specific channel being sampled.

When the frame carries TCMD information, it is only needed to address a specific node, as the status of all TCMDs on that node can be addressed by four bytes of the data field. Each TCMD then occupies two bits in the data field with the following meaning:

00 Reserved

01 Set TCMD

10 Reset TCMD

11 Leave TCMD unchanged

If one remembers that the state of each TCMD must first be assessed before it can be altered, this effective scheme has the advantage that a single TCMD on a node can be targeted, leaving all others unchanged.

7.3.2 Busloading

In any network application, it is important to know what the predicted busload would be in order to avoid overloads. This is not an easy task for this specific C&DH application. TCMDs, and to a lesser extent TLM signals, are sporadic as opposed to periodic. Furthermore, it is difficult to predict the behaviour of the system accurately before the supporting software has not been completed. Another approach is therefore taken in this section.

The maximum continuous TLM rate for all TLM channels is calculated in order to realise a 60% busload at different bus speeds. This is done under the assumption that the periodicity of TCMD packets is much less than that of TLM, and that all TLM channels have the same period. The resultant busload is then tested for overload by adding a short burst of TCMDs.

The transmission time, C_m, of a TLM frame can be worked out using Equation 7.2. All TLM frames contain eight data bytes and the bus is running at 125Kbps. To calculate the total busload, the load caused by each of the 160 TLM channels must be added up. The following equation can then be re-arranged in order to obtain a value for T_m, the period of message *m*, in order to realise a 60% busload:

$$N_{channels} * 100\% * \frac{C_m}{T_m} \leq 60\% \qquad \text{Equation 7.6}$$

Substituting values for $N_{channels}$ and C_m, Equation 7.6 simplifies to:

$$T_m \geq 289\text{ms} \qquad \text{Equation 7.7}$$

Taking into account that each TLM frame contains four TLM samples, the effective maximum sustainable real-time TLM rate of a specific channel is 13.84Hz (using Equation 7.7). To work out the additional busload for a burst of TCMDs, C_m, is calculated for a six-byte CAN frame. The answer is then divided by 5ms, the assumed frame interval within the burst, to obtain the resulting busload. Therefore,

$$\text{Load}_{(TCMD)} = 100 * \frac{0.924}{5} = 18.48\% \qquad \text{Equation 7.8}$$

Table 7.4 summarises the above calculations for a range of different bus speeds. The TCMD column indicates the additional busload incurred during a burst of ten TCMD packets. The dark shaded cells correspond to the undesirable condition of a total busload above 90%. On SUNSAT-1, the maximum TLM sampling rate per channel is 3.13Hz. Comparing this value to the TLM columns in the table, it can be seen that this sampling rate is exceeded even when the busload is only 20% and the bus is running at 125Kbps.

Having an acceptable busload level does not necessarily mean that all messages can be scheduled. In order to prove schedulability, it is necessary to know what the deadline for each message is. Only then can a thorough analysis be done, using Equation 7.1 and taking into account the queueing jitter and queueing delays for each message. This task is left for future research when, in addition, the nature of all messages should be known.

Table 7.4 Effective TLM sampling rate at different busloads and speeds

	1Mbps		500Kbps		250Kbps		125Kbps		9.6Kbps	
Busoad (%)	**TLM** (Hz)	**TCMD** (%)	**TLM** (Hz)	**TCMD** (%)	**TLM** (Hz)	**TCMD** (%)	**TLM** (Hz)	**TCMD** (%)	**TLM** (Hz)	**TCMD** (%)
10	18.5	2.3	9.2	4.6	4.6	9.2	2.3	18.5	0.2	240.6
20	36.9	2.3	18.5	4.6	9.2	9.2	4.6	18.5	0.4	240.6
40	73.8	2.3	36.9	4.6	18.5	9.2	9.2	18.5	0.7	240.6
60	110.7	2.3	55.4	4.6	27.7	9.2	13.8	18.5	1.1	240.6
80	147.6	2.3	73.8	4.6	36.9	9.2	18.5	18.5	1.4	240.6

7.4 On future C&DH nodes

The prototype developed for this research work includes various components and some functionality which is not needed for a flight C&DH node. A prediction is therefore made in terms of future PCB size and current consumption for a possible C&DH flight HW node. Only discrete components are used for the prediction in order to obtain the maximum possible PCB size, instead of using one or more programmable logic devices. Provision is made for 16 TCMD switches and 16 TLM channels on the modified node. Figure 7.8 shows the top and bottom silk screen plots of the four-layered PCB; and Table 7.5 shows the predicted current consumption of the components used in the modified design. For the PCB layout, surface mount components have been used as far as possible and have been placed on both sides of the PCB (see Appendix B for the schematic used).

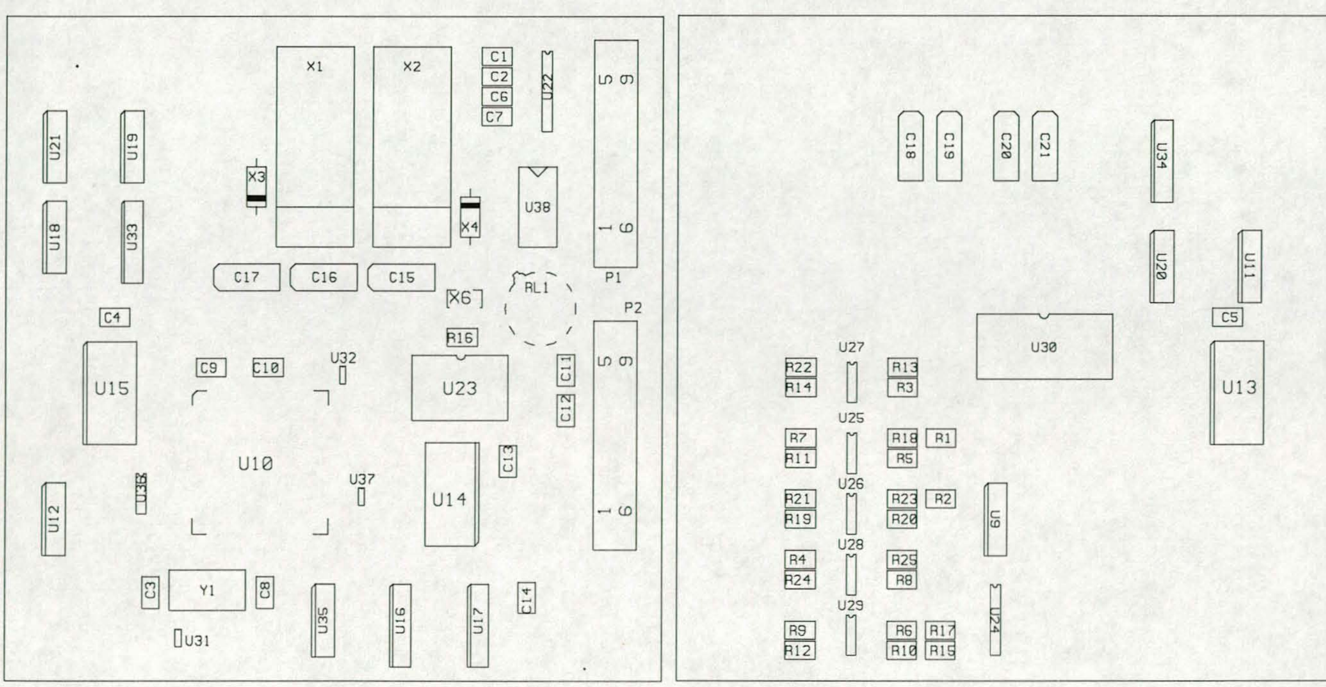

Figure 7.8 Top and bottom silkscreen plots of a sample flight C&DH node (actual size shown)

Table 7.5 Predicted current consumption on a flight C&DH node

Component	Count	*I_{Active} (Single)	*$I_{low\ power}$ (Single)	*I_{Active} (Total)	*$I_{low\ power}$ (Total)
C515C @ 10MHz	1	18.92	50μ	18.92	50μ
28F001	1	13	30μ	13	30μ
628128	1	15	2μ	-	2μ
Max242	1	15	2μ	-	2μ
Max875	1	2	2	2	2
HC4051	2	80μ	80μ	160μ	160μ
HC573	3	80μ	80μ	240μ	240μ
HC164	2	80μ	80μ	160μ	160μ
HC251	2	80μ	80μ	160μ	160μ
HC86	4	20μ	20μ	80μ	80μ
HC08	2x Single	10μ	10μ	20μ	20μ
HC04	2x Single	10μ	10μ	20μ	20μ
HC32	1	20μ	20μ	20μ	20μ
HC126	1	4μ	4μ	4μ	4μ
HCPL-0630	5	11	11	55	55
TLE6252	1	6	15μ	6	15μ
82C250	1	15	170μ	-	170μ
*All values in mA unless otherwise indicated			**Total:**	96mA	58mA

Note that the largest contributor to the current consumption is the opto-couplers. When a high quality shielded, twisted-pair cable is used for the CAN bus and no galvanic isolation is provided, the current consumption can be reduced to 41mA and 3mA in normal and power down modes respectively. If this is the case, then the overall C&DH system, consisting of for instance,10 nodes, will consume about 492mA and 36mA respectively[8] for the two operational states. But when opto-couplers are use, the default system current consumption rises to 1152mA[8].

7.5 Conclusions

This chapter provided insight into various practical aspects surrounding the proposed C&DH node architecture and the prototype board. In particular, the fault tolerant capabilities of the prototype board were highlighted, as well as the flexible nature of the

[8] A 20% margin is added to make provision for unaccounted current consumption.

design. It does indeed conform to the principles of physical modularity and electronic flexibility considered in Chapter 3. In addition, it was shown that at a relatively low bus speed of 125Kbps, the effective sampling rate of all TLM channels (at a busload of 60%) is still more than four times better than the highest TLM sampling rate on SUNSAT-1.

From the evaluation of the fault tolerant CAN transceiver (TLE6252), it is evident that the 125Kbps bus speed is a limiting factor for this particular C&DH design. However, it was theoretically shown that relatively high TLM sampling speeds can be sustained with this transceiver. Even when adding short bursts of TCMD on a 60% loaded bus, the bus does not become overloaded. In future however, the fault tolerant transceiver may prove to be ineffective when more that ten nodes are incorporated into the C&DH design.

Power consumption on the C&DH system remains a critical issue. Depending on the final SW and sampling needs, the processors on some or all nodes may run at lower than 10MHz clock speeds. This will save additional power, given the linear relationship between processor clock speed and power consumption. However, a lower clock speed will also limit the maximum speed on the CAN bus. For example, a C515C CPU running at 2MHz can only sustain a maximum CAN speed of 250Kbps. Furthermore, the processor can also be switched into an idle mode or a slow-down mode for intermediate power consumption states.

Chapter 8 Future research

The bulk of the research presented in this document covers only the most important aspects of a basic C&DH design for future SUNSAT satellites. Some of the shortfalls identified during the project are given here as topics for future research.

The CAN protocol implements only the lowest two layers of the ISO network reference model in hardware. Software has to be written in order to implement some or all of the higher layers, in particular layer 7, the application layer. In CAN terminology, this is called a higher layer protocol (HLP). Several complete and ready to implement HLPs do exist on the market, for instance CAN Application Layer (CAL), CANOpen, CANKingdom, DeviceNet and others. Research in this area will therefore include an investigation into the viability of adaptation of one of these HLPs for a satellite application. On the other hand, the merits of writing a custom HLP aimed at the specific requirements of the C&DH system require evaluation.

The CCSDS recommendations on TLM and TCMD must also be adapted into a usable form for an uplink and downlink protocol on future satellites. Surrey Satellite Technology Limited (SSTL) in the United Kingdom has been developing micro- and mini-satellites for more than a decade, and although they have recently started to use CAN technology in space, they still use a long standing RF protocol for uplinking TCMDs and downlinking TLM. As stated before, it is expected that the CCSDS protocols will enhance the inter-operability of future spacecraft and therefore continued research in this area is merited.

Once the requirements and specifications for a future C&DH system have been finalised, the clock speed for the nodes has to be determined. This will not only dictate the current consumption of the network, but also the bus speed and therefore the busload as well as the schedulability of CAN messages on the system.

The Infinion processor used in the design of the prototype C&DH system, has been found to be easy to implement and effective in terms of on-board peripherals and processor power. However, new devices on the market, such as a Dallas 8051-derivative with two on-board CAN controllers, and a Philips device with an on-board transceiver, expand the potential capabilities of a C&DH system. Consideration of these in future research could prove useful.

The prototype developed as part of this research, makes provision only for level and pulse TCMDs, and one-line analog or digital TLM channels. Depending on the final requirements of the subsystems on a future satellite, other types of TCMD and TLM interfaces may need to be added to the current design as discussed in Paragraph 2.2.3. This can include a serial TCMD stream, or a data word being latched into a register on the target subsystem. TLM sampling capabilities can be expanded by latching in 8-bit wide digital words at a time.

Other aspects which usually fall under TLM and TCMD, such as coding, encryption, authentication, signal conditioning and modulation are all topics that need to be addressed in future to complete the overall C&DH design.

Several advantages of a *distributed* C&DH architecture have been highlighted in this document. It was also shown that such a system has distinct advantages over a centralised architecture (as on SUNSAT-1). However, from the viewpoint of commercial subsystem development, a distributed subsystem architecture may prove to be difficult to market. A major problem is the production, environmental testing and delivering of a fully integrated (stand-alone) device. Although the benefits of a distributed system are clear, the focus of such a development for future use will have to be determined.

Chapter 9 Conclusions

This thesis investigated the architecture of a Command and Data Handling (C&DH) system for future SUNSAT satellites. The design of the two underlying components of such a system, namely telemetry (TLM) and telecommand (TCMD), was examined by looking at the functionality incorporated into these components to make the C&DH system complete and reliable. The design of the TLM and TCMD subsystems on SUNSAT-1 was explained in order to highlight the key shortcomings of both. This enabled the identification of suggested improvements, the most important of which were improved feedback of TCMD states on the satellite and a more flexible TLM system in order to optimise the use of bandwidth.

For increased flexibility of the TLM system, a bus architecture was proposed. The two buses currently used on SUNSAT-1 were found to be too restrictive in terms of architecture and SW overhead. *FireWire* and USB, do satisfy the requirements of the proposed C&DH system, but would over complicate it due to their intricate architecture and elaborate protocols. The category of field buses was subsequently explored, looking at Bitbus, Profibus, Lonworks and CAN. From the evaluation of these four architectures, it was shown that CAN provides a simple and low cost, though fault tolerant and reliable vehicle for relatively high speed (up to 1Mbps) communication over a single twisted-wire pair.

The structural design of a packetised TLM and TCMD system was developed by making use of several identical C&DH nodes based on CAN technology. The importance of fault tolerance was addressed by basing the design on a dual-redundant CAN network, and by making use of component-level redundancy on each C&DH node in parallel with the built-in fault tolerant capabilities of CAN. Furthermore, a degree of satellite autonomy was realised by implementing a micro processor on each node. This further extends the functional capabilities of the satellite, especially in remote parts of orbit.

A subsequent prototype implementation was limited to the design of a computer system, and data and communication interfaces. The main requirements for the design were flexibility, TCMD feedback and low power. These requirements were achieved satisfactorily. By making use of nodes in a C&DH network, it is possible to expand the system for future needs without the need to change the system architecture. This makes the design inherently flexible. Flexibility is further extended by the various operational states that can be supported on the system, for instance, different TLM sampling rates, the interchange of house keeping parameters amongst the various nodes and back-up paths for settings TCMDs. A feedback mechanism was implemented, where the state of each TCMD switch can be read back by the processor on the individual C&DH nodes.

Although several of the components on the prototype board can be switched into low power modes, the default current consumption remained relatively high at 146mA. However, when high current consuming components were subsequently omitted in a sample C&DH flight design, the predicted current consumption dropped to 41mA per node. The feasibility of using full galvanic isolation for the CAN drivers on the nodes still has to be determined, as opto-isolators significantly contribute to the total current consumption on each node.

A new fault tolerant transceiver was also evaluated as part of the prototype design, and it was shown (by measurement) that its capabilities can further improve the reliability of a CAN network. The transceiver limits the maximum bus speed to 125Kbps, but a subsequent prediction has shown that a sampling rate of 13.8Hz per TLM channel for a total of 160 channels can be sustained when the bus is 60% loaded and running at this speed. This sampling rate is almost ten times better than the fastest sampling rate on SUNSAT-1.

Apart from the prototype design, three international TLM and TCMD standards have also been evaluated, namely those of the CCSDS, ESA and IRIG. The CCSDS standards are the most widely used throughout the world. The ESA ones, derived from the CCSDS standards, are mainly used by the European Union member states. IRIG TLM standards were briefly mentioned, but their significance outside the United States DoD is very small. It was proposed that the CCSDS protocols be implemented between the satellite and ground station link in order to increase the inter-operability of future SUNSAT satellites and therefore raise the international esteem of the project. Due to the short message length of a CAN frame, the chosen CCSDS protocol should not be implemented between the various subsystems on the satellite, but only between the satellite and ground station. It was also recommended that the AX.25 protocol be included on future satellites, not only for possible radio amateur use, but also as a reliable general packet transfer protocol.

Part D

Appendices

A. CAN technology detail

B. C&DH prototype schematics

C. Frame formats of the CCSDS TLM & TCMD standards

D. List of assigned CCSDS spacecraft IDs

Appendix A — CAN technology detail

A basic understanding of the details of CAN technology is assumed in preceding chapters. The most important and relevant aspects are presented below.

A.1 The physical layer

The CAN protocol uses medium access control through message priority with nondestructive bit-by-bit contention on the transmission medium [Lawrenz, 1997:179]. Consequently, any CAN physical layer needs to support the representation of a recessive[1] and a dominant state on the transmission medium, for instance twisted wire-pairs and fibre optics. The bitwise arbitration bus access mechanism of CAN discussed above, is realised by recessive and dominant states. With twisted wire-pairs, the signalling is carried out using differential voltages and it is from this that CAN derives much of its noise immunity and fault tolerance. It is by far the most common signalling method and will also be used in the design of the new C&DH system.

A.2 CAN implementations

Since the inception of CAN technology in the early 1980s, it has taken on various forms and has also progressed through several development phases. Today, CAN implementations can be grouped into three sections or a combination thereof:

- **BasicCAN and FullCAN**
  BasicCAN has a tight coupling between the CPU and the CAN controller, where all messages broadcast on the network have to be individually checked by the microcontroller. This results in the processor being tied up checking messages rather than processing them. With FullCAN implementations, an acceptance filter masks out the irrelevant messages using the message IDs, and stores it in a bank of message objects which acts as a dual-ported RAM between CAN controller and the CPU.

[1] The two lines in the twisted pair are termed CAN_H and CAN_L and both are at a level of 2.5V in the quiescent state. A '0', or logical low, is denoted by CAN_H being at a voltage level of at least 1.5V higher than CAN_L and as such is termed a dominant bit. A recessive bit ('1') is realised when the voltage of CAN_L is not higher than the voltage of CAN_H plus 0.5V. A dominant bit therefore overrides a recessive bit on the communication medium.

- **Standard CAN and Extended CAN**
  These two terms refer to the length of the CAN message identifier. As prescribed in the CAN Specification 2.0A, the so-called 'standard' CAN frames provide an 11-bit identifier, while CAN Specification 2.0B defines a 29-bit long identifier, representing the so-called 'extended' CAN frames. In order to avoid problems in a mixed network, some CAN implementations provide a CAN 2.0B 'passive' feature, where all 29-bit ID messages received by such a node will simply be ignored.
- **Stand Alone CAN and Integrated CAN**
  The very first CAN silicon was known as a 'stand alone' version: it did not have any CPU or driver functionality, and implemented only the CAN protocol. In following years, various processor integrated CAN solutions appeared on the market and were subsequently termed 'integrated CANs'. The former leaves the choice of processor to the designer, but the latter provides the advantage of higher reliability and less cost.

A.3 CAN frame formats

Every CAN message consists of a certain number of bits, divided into six fields: arbitration, control, data, CRC, ACK and end of frame fields. The arbitration field includes the priority of the message and the message ID. This identification word consists of 11 bits for CAN 2.0A (standard) messages, and 29 bits for CAN 2.0B (extended) messages. Figure A.2 at the end of this chapter shows a standard CAN frame and Figure A.3 shows the arbitration field of an extended frame. This is the only field that differs from standard to extended CAN frames.

As described in the ISO-11898 specification document [ISO 11898, 1993], 2032 different message IDs[2] are allowed for CAN 2.0A. In extended CAN networks, 2^{29} different messages are permitted. The meaning of the bit fields for Standard CAN messages are as follows:

- **Start bit** (1 bit = dominant)
  Marks the start of a message. After an idle time (bus not used), the falling edge of the start bit is used for synchronisation of the different network nodes.
- **Identifier** (11 bits)
  Contains the priority and ID of the message. The lower the value, the higher the

[2] The limit of 2032 (and not 2^{11} = 2048) can be traced back to an implementation restriction of the Intel 82526 CAN controller. The original Bosch CAN specification took this into account as did the ISO 11898 specification. Most controllers today though, can address 2048 different messages.

priority (0 = highest priority). The MSB is transmitted first.

- **RTR bit** (1 bit)
  The Remote Transmission Request bit is used by a receiver to request a remote transmitter to send his information (RTR = '1'). In this case the data field is empty. The request and subsequent answer are two completely different frames on the bus. This means the answer can be delayed due to messages with higher priorities.
- **IDE bit** (1 bit)
  If this bit is dominant, it means that the present frame is a standard CAN frame.
- **r0** (1 bit)
  Reserved.
- **Data length code** (4 bits)
  Sets the length of the following data bits. Valid values are 0 to 8.
- **Data field** (0 to 64 bits)
  Contains the application data (if any) of the message.
- **CRC sequence** (15 bits)
  Contains the checksum for the preceding bits of the message. The CRC checksum will be used for error detection only. It is not used for error correction. The Hamming Distance of the code is 6. With this, it is possible to detect up to 5 single bit errors or burst-errors up to a length of 15 bits.
- **CRC delimiter** (1 bit)
  This bit directly follows the CRC sequence, and must be recessive.
- **Acknowledge slot** (1 bit)
  All nodes in the network which have received the CRC sequence report this within the ACK slot (first bit of the ACK field) by superscribing the recessive bit of the transmitter with a dominant bit.
- **Acknowledge delimiter** (1 bit)
  This is the second bit of the ACK field, and must be recessive.
- **End of frame** (7 bits)
  Each data frame and remote frame is delimited by a flag sequence consisting of seven recessive bits.

The extended CAN messages differ from standard CAN messages in the following positions:

- **SRR** (1 bit = recessive)
  The Substitute Remote Request (SRR) bit is transmitted at the position of the RTR bit in standard frames.

- **IDE** (1 bit = recessive)
  In extended CAN messages, the IDE bit is recessive to indicate that more ID bits will follow.
- **Control field** (6 bits)
  The first two bits in the control field (r0 & r1) of an extended frame are reserved. The rest of the field has the same meaning as in standard CAN messages.

A.4 Frame Coding

The frame segments *start bit*, *arbitration field*, *control field*, and *CRC sequence* are coded with a method of bit stuffing. Whenever a transmitter detects five consecutive bits of identical value in the bitstream, including stuff bits, it automatically inserts a complementary bit in the bitstream actually being transmitted. The remaining fields (CRC delimiter, ACK field and EOF) are of fixed form, and are not stuffed. The bitstream is coded according to the NRZ (Non-Return-to-Zero) method. This means that the generated bit level is constant during the total bit time.

A.5 Exception handling

All errors on the CAN network are handled by the built-in error-management unit of the CAN controller. This forms part of the data link layer - layer 2 of the ISO network reference model. Every error detected by a network node will be notified to the rest of the nodes immediately. After this error message, all the nodes discard the bits received. The transmitter of the message will stop transmitting when detecting opposite bits on the network to the actual bits transmitted, and will repeat the message when the bus is idle again.

A.5.1 Error frame format

The error frame shown in Figure A.4, consists of two different fields. The first field is given by the superposition of error flags contributed by those nodes which detected the error on the bus. The second field is the error delimiter. There are two forms of an error flag: an active error flag, and a passive error flag. An active error flag consists of six consecutive dominant bits, and may only be transmitted by error active nodes (see Error Limitation below). A passive error flag consists of six consecutive recessive bits, unless it is overwritten by dominant bits from other nodes. Passive error flags are transmitted by error passive nodes.

An active error flag either violates the rule of bit stuffing, or destroys the fixed form ACK or EOF fields. As a consequence, all other nodes detect an error condition and begin transmission of an error flag. The sequence of dominant bits which can be monitored on the bus results from a superposition of different error flags transmitted by individual nodes. The total length of this sequence varies between a minimum of six, and a maximum of 12 bits. The error delimiter consists of eight recessive bits. After transmission of an error flag, each station sends recessive bits and monitors the bus until it detects a recessive bit. Afterwards, it starts transmitting seven additional recessive bits.

The error process is divided into three parts: error detection, error handling and error limitation [Lawrenz, 1997:89]:

A.5.2 Error detection

The error management unit has five different built-in error checks:

Message-level error detection

- **Cyclic Redundancy Check** (CRC)
  The CRC safeguards the information in the frame by adding redundant check bits at the transmission end. At the receiver end, these bits are re-computed and tested against the received bits. If they do not agree, there has been a CRC error.
- **Frame check**
  This mechanism verifies the structure of the transmitted frame by checking the bit fields against the fixed format and the frame size. Errors detected by frame checks are designated format errors.
- **ACK errors**
  If no acknowledgement is received by the transmitter of a message (ACK error), this may mean one of three things. There is either a transmission error which has been detected only by the recipients; the ACK field has been corrupted; or there are no receivers.

Bit-level error detection

- **Bit monitoring**
  A node that is sending a bit on the bus also monitors the bus. A bit error is detected when the bit value that is monitored differs from the bit value sent.
- **Bit stuffing**
  A stuff error is detected when six consecutive equal bit levels occur in a frame field that should be coded by the method of bit stuffing.

A.5.3 Error Handling

Error handling occurs in the following order:

1. An error is detected - any of the five described above.
2. An error frame will be transmitted in the form described under A.5.1.
3. The message will be discarded by every network node.
4. The error counters of every bus node will be incremented (see A.5.4).
5. The message transmission will be repeated.

A.5.4 Error limitation

To prevent a permanently disturbed bus due to error frames caused by a local disturbance of one or a group of network nodes, a special algorithm is implemented to limit the effect of this kind of errors. Each CAN controller has three error states.

- **Error Active**

  An error active node can usually take part in bus communication and send an active error flag when an error has been detected.

- **Error Passive**

  An error passive node can normally take part in bus communication and send a passive error flag when an error has been detected. After transmission, an error passive node will wait for an additional period of time before initiating a further transmission.

- **Bus Off**

  A node is in bus off state when it is switched off from the bus due to the request of a fault confinement entity. In the bus off state, a node can neither send nor receive any frames.

A state machine is implemented in the CAN controller to switch the error state of a node between the different states mentioned above. A node can only leave the bus off state upon user request (hardware or software reset). The different states depend on the values of the internal error counters. One Receive Error Counter (REC), and one Transmit Error Counter (TEC) are implemented for each node. Figure A.1 shows the state diagram for the CAN error management unit. The error counters are modified according to a complex set or rules (see [ISO 11898:1993] for details).

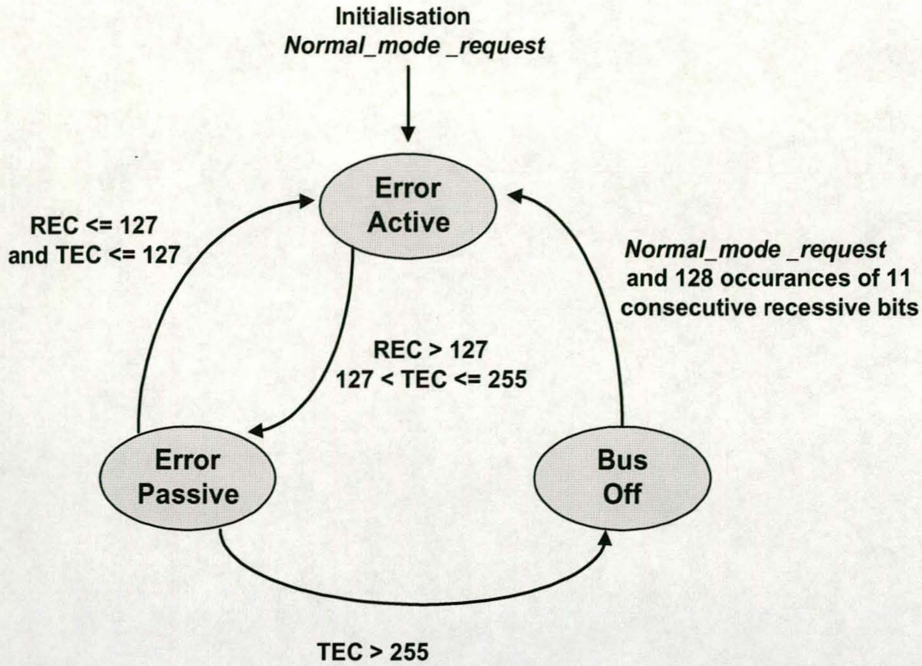

Figure A.1 CAN error states
Source: Lawrenz [1997:91]

A.6 Timing considerations

As described above, there are two different frame types: standard CAN messages, and extended CAN messages. The number of data bytes can range from zero to eight bytes. Due to this, the data rate and the delay times depend on the type of frame and the data length. The maximum delay time for a message with the highest priority depends on the duration of the message with the maximum length and the transmission rate. In a standard CAN network, this time is calculated as follows:

	1	Start bit
+	11	Identifier bits
+	1	RTR bit
+	6	Control bits
+	64	Data bits
+	15	CRC bits
+	24	(maximum) stuff bits
+	1	CRC delimiter
+	1	ACK slot
+	1	ACK delimiter
+	7	EOF bits

+	3	IFS bits
=	135	bits

The maximum delay time for a bus access of the above message is therefore 130 bit times (e.g. 135μs with a maximum transmission rate of 1Mbit/s).

In an extended CAN network, the time is calculated as follows:

	1	Start bit
+	11	Identifier bits
+	1	SRR bit
+	1	IDE bit
+	18	Identifier bits
+	1	RTR bit
+	6	Control bits
+	64	Data bits
+	15	CRC bits
+	29	(maximum) stuff bits
+	1	CRC delimiter
+	1	ACK slot
+	1	ACK delimiter
+	7	EOF bits
+	3	IFS bits
=	160	bits

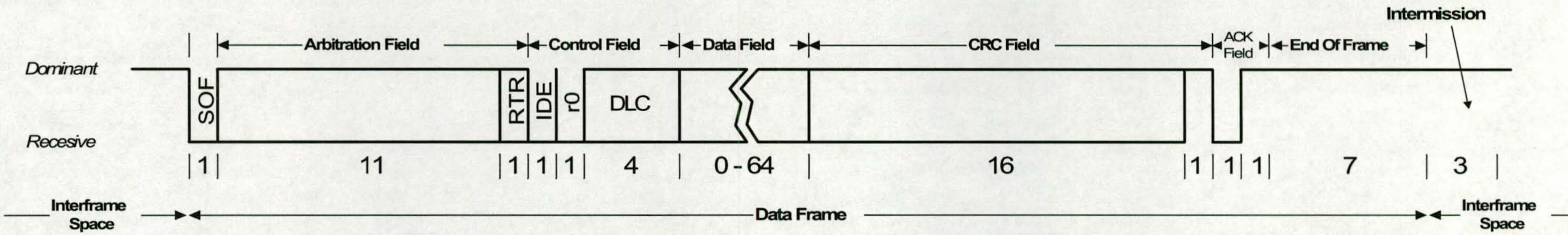

Figure A.2 Standard CAN frame format

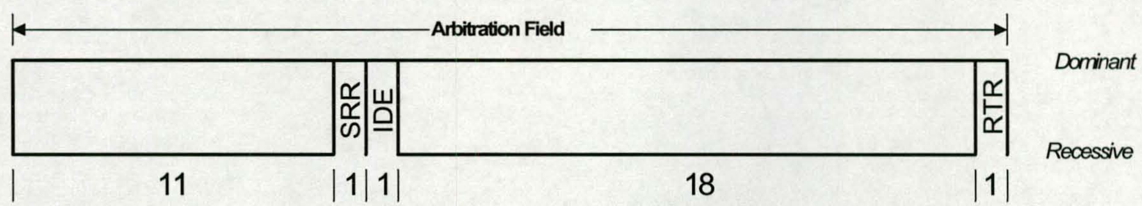

Figure A.3 Arbitration field of extended CAN frame

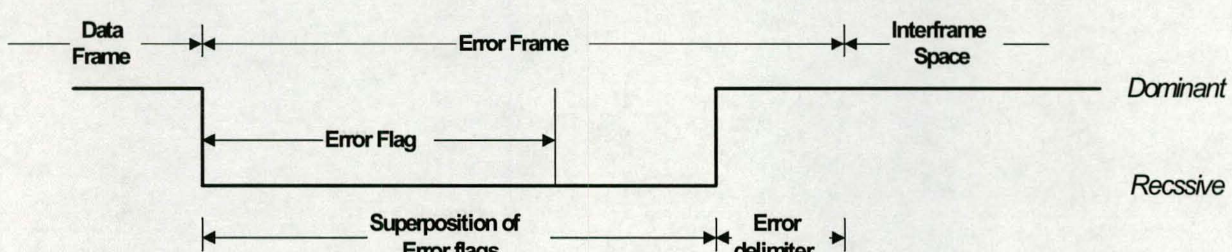

Figure A.4 CAN error frame

Appendix B

C&DH schematics

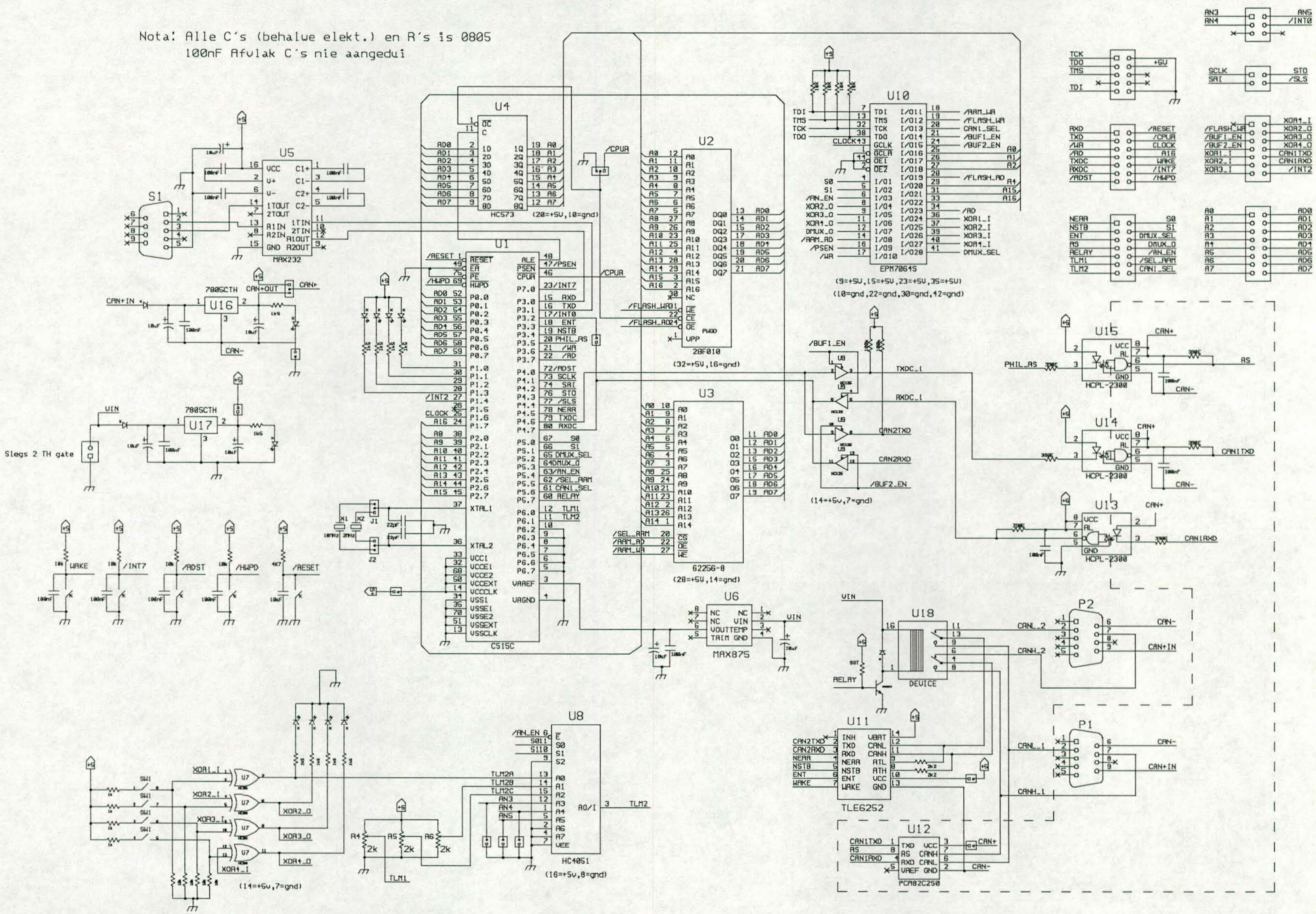

Figure B.1 C&DH node prototype board schematics

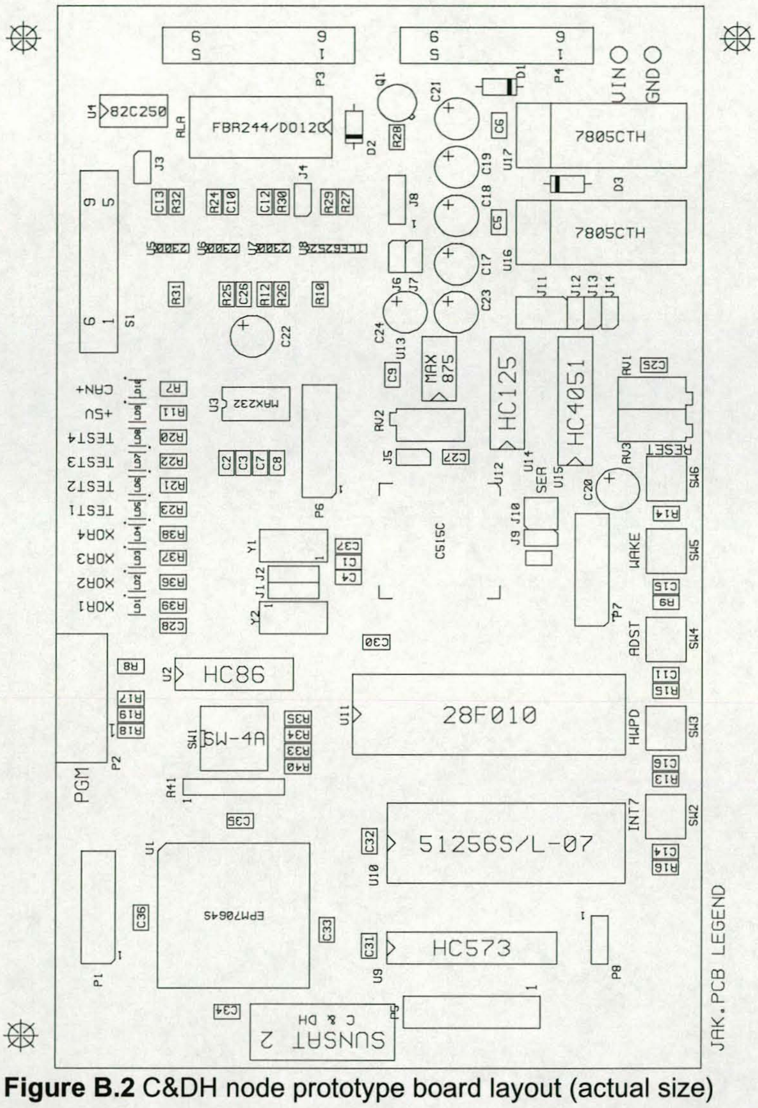

Figure B.2 C&DH node prototype board layout (actual size)

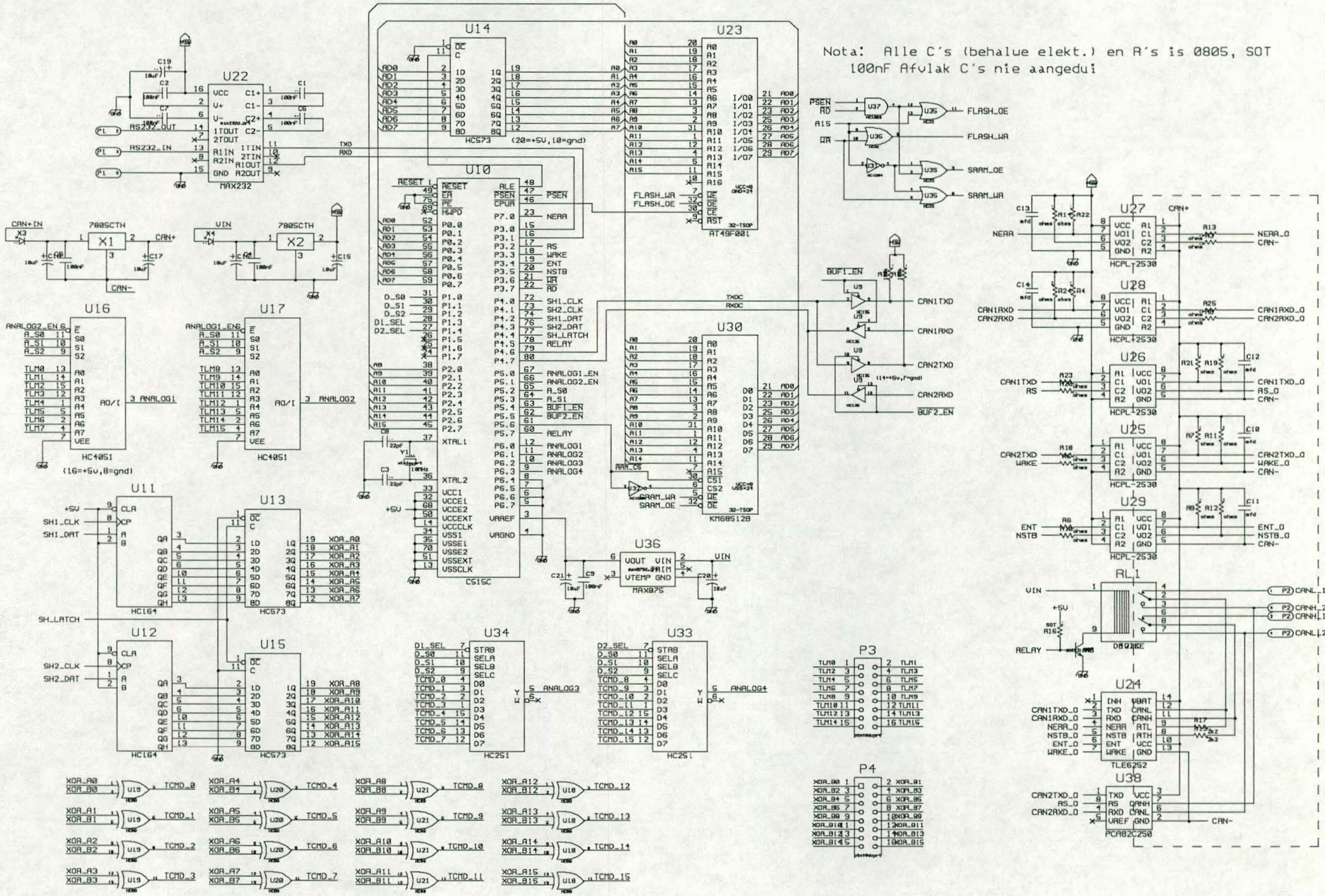

Figure B.3 Schematics of C&DH sample flight node

Appendix C

Frame formats of the CCSDS TLM & TCMD standards

This section summarises the frame formats used in the CCSDS recommendations on packet telemetry and telecommand. For full details, refer to the individual CCSDS reports quoted in the sections below.

C.1 Packet telemetry

"The essence of the packet telemetry concept is to permit multiple application processes running in on-board sources to create units of data [...] and then to permit the on-board data system to transmit these data units over a space-to-ground communications channel in a way that enables the ground system to recover the individual data units with high reliability and provide them to the data sinks in sequence" [CCSDS, 1995]. Two basic data structures - the source packet and transfer frame - are identified to accomplish these functions.

C.1.1 The source packet

Each TLM application process on-board the satellite generates a data structure called a source packet. The definition of an application process is any entity on a satellite subsystem that generates TLM data, and which is connected to a C&DH node. The internal data content of each source packet is under complete control of the application process, may be of fixed or variable length, and may be generated at fixed or variable intervals.

Figure D.1 shows the structure of a source packet. It contains two major fields: the primary header and data field. It encapsulates a block of data to be transmitted from the application process in space to one or several sink processes on the ground. The source packet consists of at least seven and at most 65542 bytes. The **version number** is a fixed 3-bit field containing the value "000". A **type indicator** field distinguishes the source packet from a similar telecommand structure and is fixed at "0". If the source packet contains a secondary header, the **packet secondary header flag** is set to "1". Otherwise it is set to "0". An 11-bit **application process identifier** uniquely identifies the source of the data and is different for different application processes.

The packet sequence control field provides a sequential count of the packets generated with the same application process identifier and, if the grouping feature is applied, provides information on the position of a source packet in the group. The 2-bit **grouping flag** has the meaning indicated in Figure D.1. A continuous modulo-16384 **source sequence count** provides a sequential binary count of each source packet generated by an application process.

The contents of the 16-bit **packet data length** field may be variable, and allow for a maximum length of 65536 data bytes in the packet data field. The purpose of the **packet secondary header** is to allow a CCSDS-defined means for placing ancillary data within a source packet. This optional field (if a source data field is present) consists of either a data field, or a time code field, or both. If a time code field is present, it should conform to one of the CCSDS segmented binary or unsegmented binary time codes, specified in the time code formats recommendation [CCSDS, 1990]. The last field in the source packet, the **source data field**, is mandatory if no secondary header is present; if not, it is optional. The length of this field is variable.

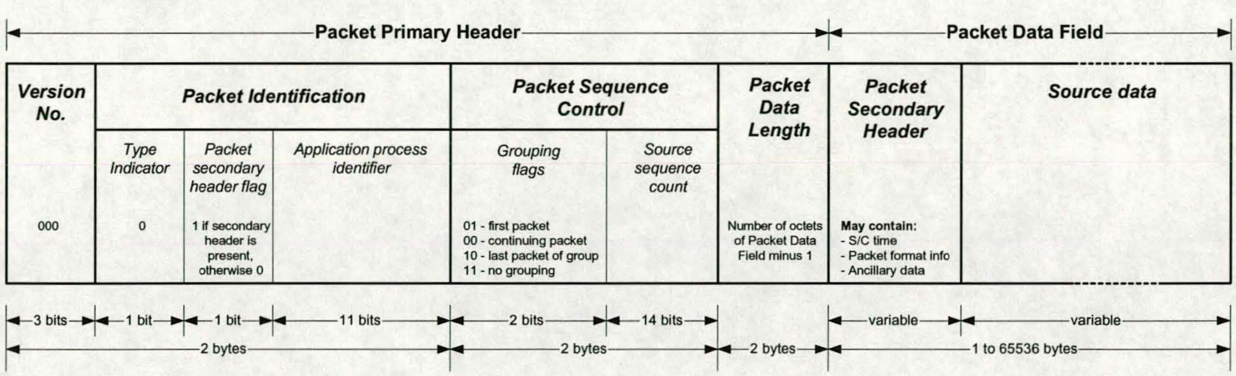

Figure C.1 CCSDS telemetry source packet

C.1.2 The transfer frame

Multiple, asynchronous application processes on a satellite can generate variable-length source packets at different rates. These source packets can then be multiplexed into a synchronous stream of fixed-length coded transfer frames for reliable transmission to the ground. The transfer frames carry information in the primary header that permits the ground system to route them to their intended destination.

The structure of a transfer frame is shown in Figure D.2. It consists of four major fields: the primary header (6 bytes), the secondary header (up to 65 bytes), the data field of variable length and two optional fields (6 bytes). The length of the transfer frame is constant throughout a specific mission, and the maximum length thereof is limited to 1115 bytes.

A **version number** identifies this data unit as a transfer frame and is set to "00". The 10-bit **spacecraft identifier** is assigned by the CCSDS (see Appendix D for currently used IDs). The following **virtual channel ID** is three bits long (see Chapter 6 for a description of a virtual channel). If the **operational control field flag** is set, it signals the presence of the operational control field at the end of the frame. If this optional field is omitted, the flag is set to "0". Each transfer frame transmitted is counted by the modulo-256 **master channel frame counter**. Similarly, each transfer frame transmitted through a specific virtual channel is counted by the **virtual channel frame count**.

If a secondary header is present, the **transfer frame secondary header flag** is set to "1", and otherwise to "0". The **synchronisation flag** signals the presence of TLM source packets within the transfer frame data field ("0") or privately defined data ("1"). If the synchronisation flag is set to "0", the **packet order flag** is reserved for future use by the CCSDS and is set to "0", otherwise the packet order flag is undefined. Similarly, if the synchronisation flag is set to "0", the **segment length identifier** is set to "11". Otherwise it is undefined. When the transfer frame data field contains source packets (synchronisation flag = "0"), the **first header pointer** indicates the location of the first source packet (in byte boundaries, starting at 0) within the data field. Otherwise the first header pointer is undefined.

If the secondary header is present - signalled by the secondary header flag - the **secondary header version number** is set to "00". In this case, the **secondary header data length** can be used to compute the location of the start of the frame data field. The **secondary header data field** is of variable length and can extend up to 63 bytes. Source packets are inserted continuously and in forward order into the transfer frame data field. The length of this field is constrained by the length of the total transfer frame, which is fixed for a specific mission.

The first two bits of the optional **operational control field data** have the following special meaning: the first bit is used to distinguish between two different reports. If set, a type-1 report follows; otherwise it is a type-2 report. A type-1 report contains a command link control word (CLCW) - see the data routing service recommendation [CCSDS, 1987c] for details of this status reporting mechanism within the transfer layer of the packet TCMD structure. The second bit indicates whether a type-2 report contains project-specific data ("0"), or if the contents are reserved by the CCSDS for future applications ("1"). The last field, **frame error control field data**, is only optional if the transfer frame is Reed-Solomon encoded, otherwise it is mandatory. In the latter case, a systematic binary (n,n-16) block code is appended to the end of the transfer frame.

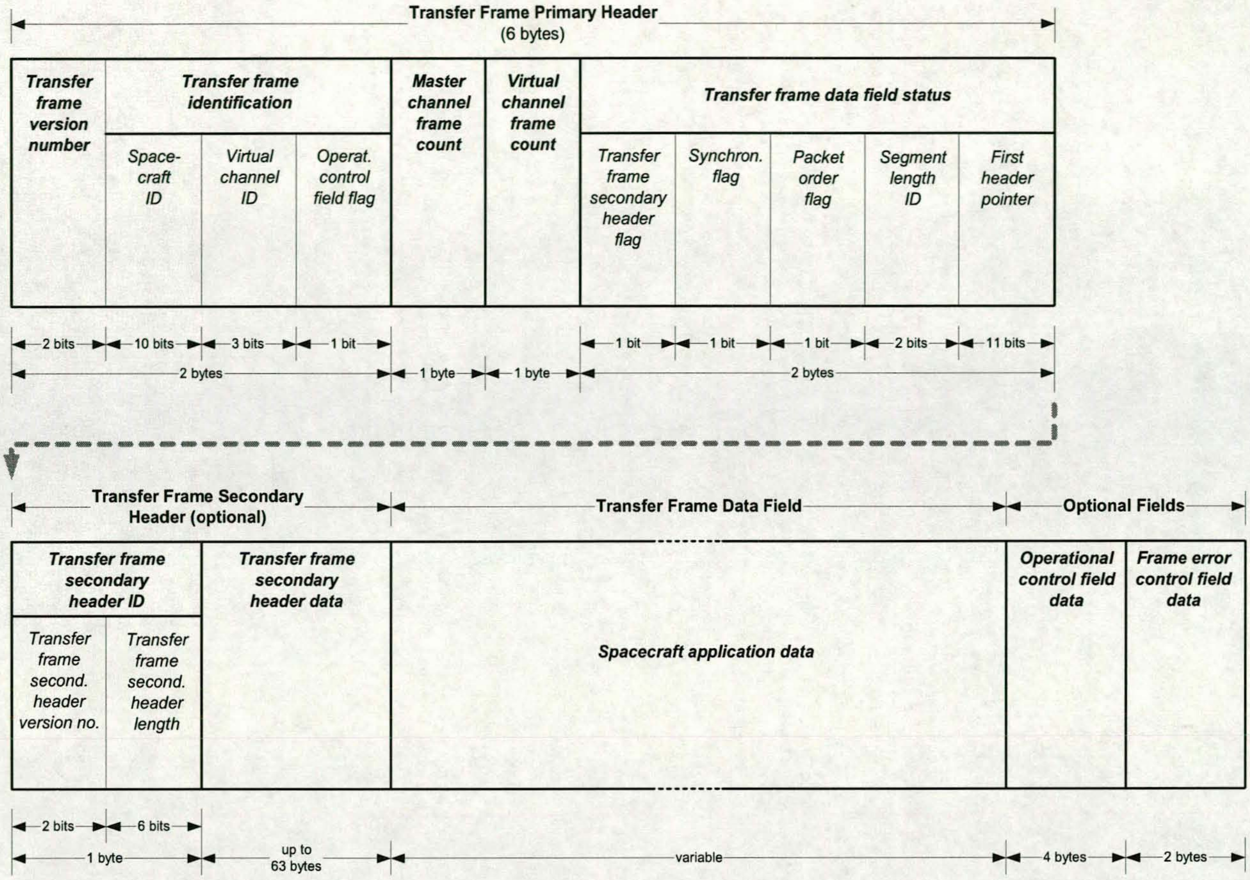

Figure C.2 CCSDS packet telemetry transfer frame

C.2 Packet telecommand

Although the TCMD recommendations of the CCSDS are more comprehensive than those pertaining to TLM systems, two similar structures exist in the layered TCMD architecture in order to realise a packet TCMD system: the TCMD packet, and transfer frame.

C.2.1 The telecommand packet

"A [telecommand] packet is a basic user data unit that is transported 'up' to the spacecraft by the [ground] telecommand system: it is virtually identical to a telemetry packet, which is the basic user measurement data unit that is transferred 'down' to the user through the CCSDS telemetry system" [CCSDS, 1987b]. Figure D.3 shows the format of a CCSDS TCMD packet.

The three most significant bits of the packet form the **version number**, which is set to "000". The **type indicator** bit is set to "1" to indicate that this is a TCMD packet, rather than a TLM packet. If a secondary header is present, the **secondary header flag** is set to "1". If not, it is set to "0". An 11-bit **application process identifier** field uniquely identifies the receiving application process on the spacecraft. When a higher layer data structure, such as a memory load sequence, is used on an application, the **sequence flags** can be used to indicate whether the current packet is the first, last, or intermediate component of the sequence (see Figure D.3). A 14-bit **packet name or sequence count** allows a particular TCMD packet to be identified with respect to others occurring within a TCMD session. The length of the remainder of the TCMD packet is indicated by the contents of the16-bit **packet length** field. This field therefore dictates the maximum length of a TCMD packet to be 65542 bytes. A **secondary header** may contain ancillary data that can be used for the interpretation of the application data following this field. The format of its contents is not specified, as long as the first bit of the field is used to indicate whether the contents are non-CCSDS defined ("0"), or reserved for future defined CCSDS data ("1"). The last field in the TCMD packet contains the application data and might, at the discretion of the user, include an error-detecting polynomial to verify the integrity of the TCMD packet.

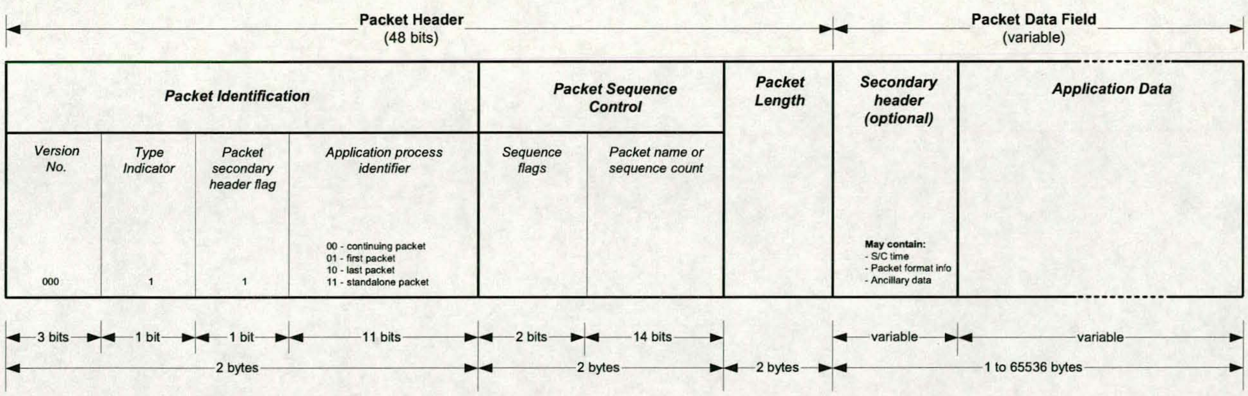

Figure C.3 CCSDS telecommand packet

C.2.2 The telecommand transfer frame

"The telecommand transfer layer [of which the transfer frame is the main data structure] is the 'heart' of the standard conventional telecommand system. It is this layer which takes care of most of the operations required to move [...] user telecommand data reliably from the sending end of the system to the receiving end in space" [CCSDS, 1987c].

Figure D.4 shows the structure of a TCMD transfer frame. Only one version of the frame is currently supported, and is indicated by setting the **version number** field to "00". The CCSDS recommendations also define a frame acceptance mechanism for TCMDs which can be bypassed by setting the **bypass flag** to "1". If the **control command flag** is set to "0", the data field of the transfer frame contains frame data such as a TCMD packet. Otherwise, the data field is used to set up parameters of the CCSDS TCMD receiver. The next two bits in the transfer frame are reserved for future use and are set to "00".

The 10-bit **spacecraft ID** is assigned by the CCSDS to uniquely identify a particular satellite. Appendix D indicates currently assigned spacecraft IDs. Up to 64 logical paths or virtual channels may be created through a single physical channel by making use of the 6-bit **virtual channel ID**, thus sharing the total transmission capacity. The total length of the transfer frame, minus one, is indicated by the **frame length** field, and allows for a total of 1024 bytes. A modulo-256 **frame sequence number** is used for sequentiality checks, and is virtual channel dependent. A separate frame sequence number is maintained for each virtual channel. The length of the **frame data field** is variable, up to a maximum of 1019 bytes (1017 if the error control field is present). If used, the optional **frame error control** field contains a 16-bit CRC word that can be used for extra error checking capability.

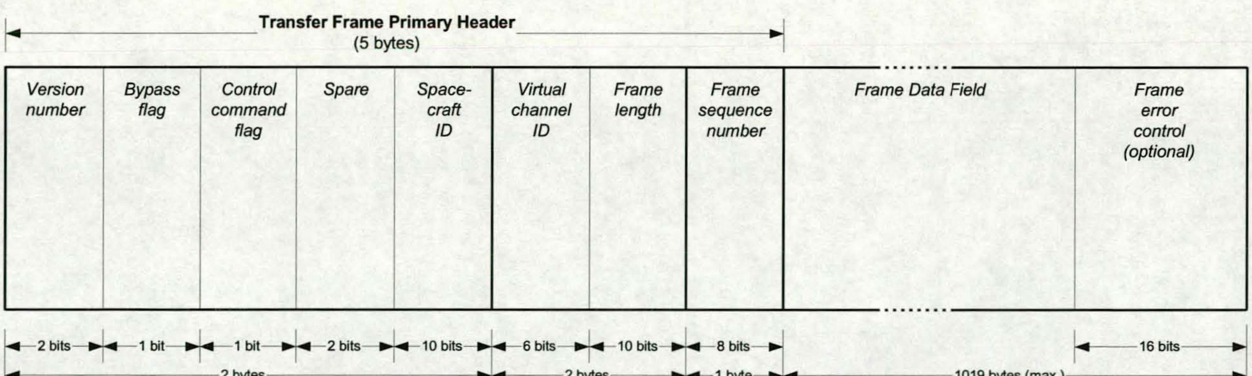

Figure C.4 CCSDS telecommand transfer frame

Appendix D

List of assigned CCSDS spacecraft IDs

```
              WORLD DATA CENTER A FOR SPACECRAFT AND ROCKETS:
      ACTIVE CCSDS SPACECRAFT ID ASSIGNMENTS FOR FRAMES / VCDUs (October 1999)

         *******************************************************
         * UNASSIGNED ID's MAY NOT BE ADOPTED BY PROJECT OFFICES.*
         *    ONLY THE WDC-A-R&S CAN ASSIGN/APPROVE CCSDS ID's,   *
         *  REQUESTED THROUGH ANY AGENCY/NATIONAL REPRESENTATIVE.*
         *******************************************************

           VERSION 1 (VN=00); SCID = 10 Bits; GSCID = VN.SCID
           ----------------------------------------------------
  COMMON NAME  !    GSCID    !GSCID!   PERSON/AFFILIATION    !ASSIGN DT! Fn
  OF SPACECRAFT!   (BINARY)  !(HEX)!   REQUESTING ID         ! INITIAL
--------------------------------!-------------------------------------!---!
Space Telescope 000000111010 !  3A ! G.M. Levin/GSFC/NASA  !              !
Nimbus 7        000000100110 !  26 ! F.   Akers/GSFC/NASA  !              !
GRO             000001001100 !  4C ! J.J. Madden/GSFC/NASA !              !
EURECA          000000101101 !  2D ! G.F. Block/ESTEC/ESA  !              !
ERS-1           000001011010 !  5A ! G.F. Block/ESTEC/ESA  !              !
Mars Observer   000010110100 !  B4 ! J.K. Erickson/JPL/NASA (1)!          !
Mars Obser(SIM) 000010110101 !  B5 ! K. Moyd/JPL/NASA      !18JAN93,RP !
ASTRO-SPAS      000000000001 !  01 ! H.   Uhrig/ESA     (2)!              !
ASTRO-SPAS Sim. 000000000010 !  02 ! H.   Uhrig/ESA        !              !
ISO             000010001101 !  8D ! H.   Uhrig/ESA     (2)!              !
ISO Simulator   000010001110 !  8E ! H.   Uhrig/ESA        !              !
Radarsat        000011001001 !  C9 ! W. E. Threinen/CSA    !              !
ERS-2           000000000011 !  03 ! H. Uhrig              !              !
ERS-2 Simulator 000000000100 !  04 !    "" ""              !              !
CRAF            000001010001 !  51 ! J. N. Scott/GSFC      !              !
CRAF-Simulator  000001011011 !  5B ! J. N. Scott/GSFC      !              !
Cassini         000001010010 !  52 ! J. N. Scott/GSFC      !              !
Cassini-Sim.    000001011100 !  5C ! J. N. Scott/GSFC      !              !
SOHO            000000010101 !  15 ! H. K. Uhrig/ESA       !              !
SOHO-Simulator  000000010110 !  16 !     "" ""             !              !
ARIANE 5        000000011010 !  1A ! R. Simo-Pons/CNES     !20MAR92 RP
SAMPEX          000010110000 !  B0 ! J. N. Scott/GSFC       06JUN92 RP
SAX             000010110001    B1 ! H. K. Uhrig/ESA        26JUN92 RP
SAX Simulator   000010110010    B2 ! H. K. Uhrig/ESA        26JUN92 RP
FAST            000010110011    B3 ! J.N.Scott/GSFC         20NOV92 RP
SWAS            000010110110    B6 ! J.N.Scott/GSFC  (4)    18JAN93 RP
HUYGENS         000010110111    B7 ! H.K.Uhrig/ESA   (5)    09FEB93 RP
HUYGENS         000010111001    B9 ! H.K.Uhrig/ESA   (5)    28APR93 RP
HUYGENS-Simulat 000010111000    B8 ! H.K. Uhrig/ESA  (5)    09FEB93 RP

[MESURpathfinder 000000110101    35 ! J.N.Scott/GSFC  (6)    02FEB94 RP
 MESURpf-Simulat 000001010100    54 ! J.N.SCOTT/GSFC  (7)    02FEB94 RP
   MESUR (HEX=35,54) RELINQUISHED BY BADRI YOUNES/GSFC. REASSIGNED BELOW]
MSP 01-ORB.FL   000000110101    35 ! B.Younes/GSFC          30NOV98 RP
MSP 01-ORB.SIM  000000100010    22 ! B.Younes/GSFC          30NOV98 RP
MSP 01-LAND.FL  000001010100    54 ! B. Younes/GSFC         30NOV98 RP
MSP 01-LAND.SIM 000001011001    59 ! B.Younes/GSFC          30NOV98 RP
ICESAT (TLM)    000111001110   1CE !       "                   "
HESSI  (TLM)    000010100111    A7 !       "                   "

OERSTED         000011000101    C5 ! H.K. Uhrig/ESA         12NOV93 RP
OERSTED-SIM     000011000110    C6 ! H.K.Uhrig/ESA          12NOV93 RP
ENVISAT         000011000111    C7 ! H.K.Uhrig(Reuse.) (8)  12NOV93 RP
EOS-AM-1(CTIU-1)000010101001    A9 ! J.N.Scott/GSFC  (9)    02FEB94 RP
EOS-AM-1(CTIU-2)000010101010    AA ! J.N.Scott/GSFC  (9)    02FEB94 RP
LANDSAT7(CTIU-1)000001010101    55 ! J.Deskevich/GSFC(10)   01SEP94 RP
LANDSAT7(CTIU-2)000001010110    56 ! J.Deskevich/GSFC(10)   01SEP94 RP
NEAR  (TLM,TC)  000011000100    C4 ! J.Deskevich/GSFC       30NOV94 RP
```

```
TRACE (TLM,TC)    000010001111     8F ! J.Deskovich/GSFC       14FEB95 RP
ARTEMIS           000010001110     8E ! H.Uhrig/ESA            14FEB95 RP
ARTEMIS (SIM)     000010001101     8D ! H.Uhrig/ESA            14FEB95 RP
ETS-7 (TC)        000011100111     E7 ! N. Iwasaki/NSDA        18APR95 RP
ROCSAT-1 (TC)     000001101000     68 ! J.J.Lee/NSPO           12JUL95 RP
HOT BIRD 2(TLM)   000011110000     F0 ! H. Uhrig/ESA           31JUL95 RP
HOT BIRD 3(TLM)   000011110001     F1 ! H. Uhrig/ESA           31JUL95 RP
HOT BIRD 4(TLM)   000011110010     F2 ! H. Uhrig/ESA           31JUL95 RP
HOT BIRD(SWSIM)   000011110011     F3 ! H. Uhrig/ESA           31JUL95 RP
HOT BIRD(HWSIM)   000011110100     F4 ! H. Uhrig/ESA           31JUL95 RP
XMM (TLM/TC)      000011000001     C1 ! H. Uhrig/ESA           19SEP95 RP
XMM(SIM)          000011000010     C2 ! H. Uhrig/ESA           19SEP95 RP
WIRE (TLM/TC)     000010011001     99 ! J.Deskevich/GSFC       16OCT95 RP
MTI (TLM)         000010100001     A1 ! J. Deskevich/GSFC      02JAN96 RP
MTI (TC)          000010100010     A2 ! J. Deskevich/GSFC      02JAN96 RP
MSG-1(TC/TLM)     000101000001    141 ! R. Wolf/EUMETSAT       26FEB96 RP
MSG-2(TC/TLM)     000101000010    142 ! R. Wolf/"    "         26FEB96 RP
MSG-3(TC/TLM)     000101000011    143 ! R. Wolf/"    "         26FEB96 RP
MSG-4(TC/TLM)     000101000100    144 ! R. Wolf/"    "         26FEB96 RP
AXAF-1(TC)        000000000101      5 ! J.Deskevich/GSFC       06MAR96 RP
KOMPSAT-1 (TC))   000000000110      6 ! E.Sim/Korea ARI        05JUN96 RP
FUSE (TC)         000000000111      7 ! J.Deskevich/gsfc       24JUN96 RP
GRAVITY PROBE-B   000001000111     47 ! J. Deskevich           24JUN96 RP
METOP1(TLMTC,S-)  000000001011     0B !H.Uhrig/ESA             01JUL96 RP
METOP2(TLMTC,S-)  000000001100     0C     ""                      ""
METOP3(TLMTC,S-)  000000001101     0D     ""                      ""
METOP-SIM (S-)    000000001110     0E     ""                      ""
EUTELSAT-F1(S-)   000000001111     0F     ""                      ""
EUTELSAT-F1(Ku)   000000010000     10     ""                      ""
EUTELSAT-F2(S-)   000000010001     11     ""                      ""
EUTELSAT-F2(Ku)   000000010010     12     ""                      ""
EUTELSAT-F3(S-)   000000010011     13     ""                      ""
EUTELSAT-F3(Ku)   000000010100     14     ""                      ""
EUTELSAT-SIM(S)   000000011011     1B     ""                      ""
EUTELSAT-SIM(Ku)  000000010111     17     ""                      ""
SESAT-F1(TLMTC)   000000011000     18     ""                      ""
SESAT-SIM(TLTC)   000000011001     19     ""                      ""
HOTBIRD-5(TLM)    000000011100     1C     ""                      ""
SNOE (TLM/TC)     000011010001     D1   J.Deskevich/GSFC     30SEP96 RP
SIRIUS2(TLM/TC)   000011010010     D2   E.Jabs/ESA           17OCT96 RP
SIRIUS2(SIM)      000011010011     D3   E.Jabs/ESA           17OCT96 RP
STARDUST(TLM/TC)  000000011101     1D   J.Deskevich/GSFC     24NOV96 RP
DS1 FLIGHT "      000000011110     1E        "               24NOV96 RP
TS BALLOON*       000000011111     1F   O.Cosentino/ASI      24NOV96 RP
(*Transmed Balloon)
Cluster-A         000010010000 !   90 ! E.Jabs;H.Uhrig/ESA   05DEC96 RP
Cluster-B         000010010001 !   91 !     ""  ""              "
Cluster-C         000010010010 !   92 !     ""  ""              "
Cluster-D         000010010011 !   93 !     ""  ""              "
Cluster (Spare)   000010010100 !   94 !     ""  ""              "
Cluster-Sim.-A    000010010101 !   95 !     ""  ""              "
Cluster-Sim.-B    000010010110 !   96 !     ""  ""              "
[ALL CLUSTER ID's WERE CANCELLED BY ME IN JULY 96; NOW REASSIGNED AT JABS
REQUEST BY PHONE CALL TO ME. JABS WILL SURRENDER ID's, IF CLUSTER COULD
NOT BE RESURRECTED....RP,5 DEC 96]
MARS SURVEYOR-
LANDER98TLM/TC    000001110100 !   74 ! J.Deskevich/gsfc     16DEC96 RP
MSL98 (SIM)       000000111100 !   3C !        "                "
MARS SURVEYOR-                                  "                "
ORBITOR98TLM/TC   000001111111 !   7F          "                "
MSO98 (SIM)       000001111000 !   78 !        "                "
NILESAT-1(TLMTC)  000000100111     27 ! E.Jabs/ESA           24DEC96 RP
CARIBSTAR(TLMTC)  000000101010!    2A !    "                 30JAN98 RP
AFRISTAR(TLMTC)   000000101001     29 !    "                 24DEC96 RP
ASIASTAR(TLMTC)   000000101000     28 !    "                 30JAN98 RP
   THE ID's FOR CARIBSTAR and ASIASTAR WERE INTERCHANGED ON 30DEC98,
   AT REQUEST FROM E.JABS; INITIAL ASSIGNMENTS WERE ON 24DEC96. (RP)
ST-1 (TLM/TC)     000000101011     2B !    "                 24DEC96 RP
ASTRA 2B(TLM/TC)  000000101100     2C !    "                    "
ROSETTA (TLM/TC)  000010010111     97 !    "                 10JAN97 RP
ROSETTA (SIM)     000010011000     98 !     "                   "
ARBSAT2PFM(TLMTC) 000010000111     87 ! E.Jabs/ESA           24JUL97 RP
ARBSAT2FM2(TLMTC) 000010001000     88 ! E.Jabs/ESA           22JUL97 RP
```

ARABSAT 2(SIM)	000010001001	E1	!	"	"	
EO-1 (TC)	000010001001	89	!	J.Deskevich/GSFC	15DEC97	RP
ABRIXAS(Eng)	000111100001	1E1	!	H.Wanke/DLR	04SEP97	RP
ABRIXAS(Flt)	000111100100	1E4	!	H.Wanke/DLR	04SEP97	RP
CHAMP	000111100010	1E2	!	H.Wanke/DLR	04SEP97	RP
TIMED(TLM/TC)	000111100011	1E3	!	J.Deskevich	11SEP97	RP
ETS-VIII(TC)	000011101000	E8	!	T.Mito/NASDA	24SEP97	RP
INTELSAT K-TV (TLM,TC)	000011010100	3D	!	E.Jabbs/ESA	23OCT97	RP
QUICKSCAT(both)	000010000110	86	!	J.Deskevich/GSFC	06JAN97	RP
JASON-1(TLM/TC)	000010000000	80		R.Ivarnez/CNES	29JAN97	RP
JASON-1(SIM)	000010000001	81		"	"	
COROT (TLM/TC)	000010000010	82		"	"	
COROT (SIM)	000010000011	83		"	"	
STENTOR(TLM/TC)	000010000100	84		"	"	
STENTOR(BACKUP)	000010000101	85		"	"	
EUTELSATW4(TLTC)	000001110111	77		E.Jabs/ESA	20MAR98	RP
INTEGRAL(TLM/TC)	000001111001	79		"	"	
USERS/SEM(TC)	000011001010	CA		T.Mito/NASDA	27APR98	RP
ORION F2 (TLM)	000001110011	73		E.Jabs/ESA	"	
MAP	000010100101	A5		J. Deskevich	10AUG98	RP
IMAGE	000010100110	A6		"	10AUG98	RP
ORBVIEW-4(TC)	000001110101	75		B.Younes/GSFC	02SEP98	RP
HISPASAT1C/TLMTC	000001110001	71		N.Bobrinsky/ESA	08OCT98	RP
HISPASAT(SIM)	000001110010	72		"	"	
SICRAL(TLM/TC)	000001100111	67		N.Bobrinsky/ESA	29OCT98	RP
PROBA (TLM/TC)	000001100110	66		"	24NOV98	RP
RESSAT (TLM/TC)	000001100101	65		"	25NOV98	RP
GENESIS(TLM/TC)	000000101111	2F		B.Younes/GSFC	03DEC98	RP
GENESIS(SIM)	000000110110	36		"	"	
VCL (TC)	000010101000	A8		"	14DEC98	RP
EURASIASAT(TLMTC	000001101001	69		N.Bobrinsky	24FEB99	RP
EURASIASAT(SIM)	000001110000	70		"	"	
PLUTO-KEIPER	000010101110	AE		B.Younes	01MAR99	RP
PLUTO-KEIPER,SIM	000010101111	AF		"	"	
EUROPA-ORBITER	000010011111	9F		"	"	
EUROPA-ORBIT,SIM	000010100000	A0		"	"	
SOLARPROBE,TLMTC	000010111011	BB		"	"	
SOLARPROBE, SIM	000010111101	BD		"	"	
SPACE INTFER	000010111110	BE		"	"	
SPACE INTFER,SIM	000010111111	BF		"	"	
PICASSO-CENA	000001001000	48		R.Ivarnez/CNES	11MAR99	RP
PICASSO-CENA,SIM	000001001001	49		"	"	
W1R (TLM/TC)	000001000011	43		N. Bobrinsky	04APR99	RP
NILESAT2(TLM/TC)	000001000100	44		"	"	
WORLDSTAR4(TLMC)	000001000101	45		"	"	
NEMO-A (TC,TLM)	000000111001	39		B.Younes/gsfc	11APR99	RP
NEMO-B (TC,TLM)	000001000000	40		"	"	
MITA(TLM/TC)	000101011001	159		C.Portelly/ASI	18APR99	RP
MARS-EXP-ORB	000000110111	37		N.Bobrinsky/ESA	05MAY99	RP
BEAGLE2 MARSLAND	000000111000	38		"	"	
STRV-1C(TLM,TC)	000111000000	1C0		P.Vaughan/BrNatSpCent	"	
STRV-1D(TLM,TC)	000111010000	1D0		"	"	
EOS-PM (UPLINK)	000010011010	9A		Younes/GSFC	27JUL99	RP
FEDSAT (TLM/TC)	000000110100	34		Jacobsen/CSIRO	27JUL99	RP
EOS-CHEM (TC)	000011001100	CC		Younes/gsfc	25AUG99	RP
GALEX (TC)	000011010100	D4		"	"	
TRIANA (TC,TLM)	000001001110	4E		"	"	
GRACE-A(TLM)	000000100100	24		"	26AUG99	RP
GRACE-B "	000000100101	25		"	"	
ATV 1 (TC/TLM)	000000100000	20		N.Bobrinsky/ESA	09SEP99	RP
ATV 2 "	000000100001	21		"	"	
ATV 3 "	000000100011	23		"	"	
ATV 4 "	000000110000	30		"	"	
ATV 5 "	000000110001	31		"	"	
ATV 6 "	000000110010	32		"	"	
ATV 6 "	000000110011	33		"	"	
ATV 7 "	000001000001	41		"	"	
ATV 8 "	000001000010	42		"	"	
ATV 9 "	000001010011	53		"	"	
ATV 10 "	000001010111	57		"	"	
ATV 11 "	000001011000	58		"	"	
ATV 12 "	000001100000	60		"	"	

ATV-SIM "	000001100001	61	"	"
CORIOLIS (TLM)	000011010101	D5	Younes/GSFC	22SEP99RP
ALOS (TC)	000001100011	63	Mito/NASDA	22OCT99RP

VERSION 2 (VN=01); SCID=8 Bits; GSCID = VN.SCID

Sp.St.Freedom	0100011000 !	118 !	J. N. Scott/GSFC	(3)	!27MAR92 RP
TOMS-EP-1	0100011001 !	119 !	J. N. Scott/GSFC		!06JUN92 RP
TOMS-EP-2	0100011010 !	11A !	J. N. Scott/GSFC		!06JUN92 RP
ENVISAT	0111000111 !	1C7 !	H.Uhrig/ESA	(8)	!16NOV93 RP
XTE(xray-time-exp)	0101101001 !	169 !	J. N. Scott/GSFC		!19NOV93 RP
EOS-AM-1(TM)	0100101010 !	12A !	J.N.Scott/GSFC	(9)	02FEB94 RP
LANDSAT7(TLM)	0100010101 !	115 !	J.Deskevich/GSFC	(10)	01SEP94 RP
GLOBE(TLM)	0100000011 !	103 !	J.Deskovich/GSFC		16SEP94 RP
ETS-7(TLM)	0111100111 !	1E7 !	N.Iwasaki/NSDA		18APR95 RP
ROCSAT-1 (TLM)	0101101000	168 !	J.J.Lee/NSPO		12JUL95 RP
ADEOS-2 (TLM)	0110100010	1A2 !	N.Iwasaki/NSDA		20JUL95 RP
AXAF-I (TLM)	0100000110	106 !	J.Deskevich/GSFC		20MAR96 RP
PLANET-B (TLM)	0100000111	107 !	I. Nakatani/ISAS		16APR96 RP
LUNAR-A (TLM)	0100001000	108 !	I. Nakatani/ISAS		16APR96 RP
KOMPSAT-1 (TLM)	0100001001	109 !	E. Sim/Korea ARI		05MAY96 RP
FUSE (TLM)	0100001010	10A !	J.Deskevich/gsfc		24JUN96 RP
METOP1(TLM,TC,X-)	0100001011	10B !	H. Uhrig/ESA		01JUL96 RP
METOP2(TLM,TC,-X)	0100001100	10C	""		""
METOP3(TLM,TC,-X)	0100001101	10D	""		""
METOP-SIM (X-)	0100001110	10E	""		""
METEOR 3M-1 (TLM)	0100011101	11D	O.D Sokolov RSA		15JUL96 RP
ISSA-JEM	0100001111	10F	T.Mito/NASDA		03JUN97 RP
ASTRO-E (TLM)	0100000101	105	T.Yamada/ISAS		15SEP97 RP
ETS-VIII	0111101000	1E8	T.Mito/NASDA		24SEP97 RP
MTSAT (TLM,TC)	0111010101	1D5	T.Mito/NASDA		23OCT97 RP
EO-1 (TLM)	0110001001	189	J.Deskevich		15DEC97 RP
USERS/SEM(TLM)	0111001010	1CA	T.Mito/NASDA		27APR98 RP
USERS/REM(TLM)	0111001011	1CB	"		"
ISS-COF(Col.Or.F)	0100010111	117	N.Bobrinsky/ESA		14JUL98 RP
ORBVIEW (TLM)	0101110101	175	B.Younes		02SEP98 RP
METEOR3/TOMS5(TLM)	0101110110	176	"		"
VCL(TLM)	0110101000	1A8	"		14DEC98 RP
ORBVIEW-3,TLM	0101110100	174	"		01MAR99 RP
NEMO(TLM)	0100111001	139	"		11MAR99 RP
CONTOUR	0111001000	1C8	"		27JUL99 RP
EOS-PM (DOWNLINK)	0110011010	19A	"		"
EOS-CHEM (TLM)	0111001100	1CC	"		25AUG99 RP
GALEX (TLM)	0111010100	1D4	"		"
GRACE-A (TLM)	0100100100	124	"		07AUG99 RP
GRACE-B (TLM)	0100100101	125	"		"
ALOS (TLM)	0101100011	163	Mito/NASDA		22OCT99 RP

CANCELLED ASSIGNMENTS; NOW AVAILABLE FOR NEW MISSIONS

Mission	Binary	Hex	Assignee	Date Cancelled	
ISEE-1	000001110100	74	J.L Green	5 Feb 95	(Reuse)
ISEE-2	000011101000	E8	"	"	(Reuse)
ISEE-3	000011001101	CD	"	"	(Reuse)
DE-1	000010000111	87	"	"	(Reuse)
DE-2	000000010011	13	"	"	(Reuse)
STS-3/OSS-1	000010011000	98	"	"	(Reused)
ENVISAT	000011000111	C7	H.Uhrig	16 Nov 93	(Reuse)
EO-1 (TC/TLM)	000100000001	101	J.Deskevich	15DEC97	(Reuse)
ABRIXAS/A	000001000110	46	B.Younes	22SEP99	(Reuse)
LUNARPROSPECT	000010011011	9B	"	22SEP99	(Reuse)

References

Barrenscheen J. (1998). ***"TwinCAN, A new milestone for inter-network communication".*** Proceedings of the 5th International CAN conference. San Jose. California. USA. November.

Botha T.F. (1994). ***"Design of a reliable telecommand system for the SUNSAT microsatellite".*** Project report presented in partial fulfillment of the requirements for the degree in Master in Engineering (Electronic) at the University of Stellenbosch.

CAN. (1993). **"International standard ISO 11898"**. *Road vehicles - Interchange of digital information - Controller Area Network (CAN) for high-speed communication*. First Edition. November.

CCSDS. (1987a). ***"Telemetry, summary of concept and rationale"***. CCSDS 100.0-G-1, Issue 1. Green Book. Consultative Committee for Space Data Systems. December.

CCSDS. (1987b). ***"Telecommand, summary of concept and rationale"***. CCSDS 200.0-G-6, Issue 6. Green Book. Consultative Committee for Space Data Systems. January.

CCSDS. (1987c). ***"Telecommand, Part 2: Data routing service"***. Recommendation CCSDS 202.0-B-1. Issue 1. Blue Book. Consultative Committee for Space Data Systems. January. (ISO/FDIS 12172).

CCSDS. (1987d). ***"Telecommand, Part 3: Data management service, Architectural Definition"***. Recommendation CCSDS 203.0-B-1. Issue 1. Blue Book. Consultative Committee for Space Data Systems. January. (ISO 12174:1998).

CCSDS. (1987e). ***"Telecommand, Part 1: Channel service, Architectural specification"***. Recommendation CCSDS 201.0-B-1. Issue 1. Blue Book. Consultative Committee for Space Data Systems. January. (ISO 12171:1998).

CCSDS. (1990). ***"Time Code Formats"***. Recommendation CCSDS 301.0-B-2. Blue Book. Issue 2. Washington, D.C. April. (ISO 11104:1991).

CCSDS. (1992). ***"Telemetry channel coding"***. Recommendation for Space Data

Systems Standards. CCSDS 101.0-B-3. Blue Book. Issue 3. Consultative Committee for Space Data Systems. May. (ISO 11754:1994).

CCSDS. (1995). ***"Packet Telemetry".*** Recommendation CCSDS 102.0-B-4, Issue 4, Blue Book, Consultative Committee for Space Data Systems. November. (ISO/FDIS 13419).

Compaq, DEC, IBM, Intel, Microsoft, NEC, Northen Telecom. (1996). ***"Universal Serial Bus Specification".*** Revision 1.0. January.

Crowder R.S. 1996. ***"Fieldbus Performance".*** Internal report of the Ship Star company. Newark. Detroit. USA.

Davoody H. (1998). ***"Keynote speech.* 2nd annual 1394 Developers'Conference"** . June. San Jose. California. USA

Hoffman G. and Moore D. (1995). ***"IEEE 1394: A Ubiquitous Bus".*** Proceedings of COMPCON 95. San Francisco. March.

Day M. (1999). ***"30 Years of commercial components in space: selection techniques without formal qualification".*** 13th annual AIAA/USU conference on small satellites. Utah State University. Utah. USA. August.

Day J.C. (1995). ***"Incorporating CCSDS Telemetry Standards and Philosophy on Cassini".*** Proceedings of Computing in Aerospace. San Antonio. Texas, USA. January.

De Swardt I. (1992). ***"SUNSAT Technical Specification. Section 3.4 & 5.26: Telemetry".*** Internal document, Electronic Systems Laboratory, University of Stellenbosch, August.

De Swardt I. (1994). ***"A telemetry system for SUNSAT".*** Project report presented in partial fulfillment of the requirements for the degree in Master in Engineering (Electronic) at the University of Stellenbosch.

Electronics World. (1999). ***"DataUpdate".*** Published in Volume 105, Number 1763 of

Electronics World". November.

ESA. (1988). ***"Packet telemetry standard"***. ESA PSS-04-106, Issue 1. European Space Agency. January.

ESA. (1989). ***"Telemetry channel coding standard"***. ESA PSS-04-103, Issue 1. European Space Agency. September.

ESA. (1992). ***"Packet telecommand standard"***. ESA PSS-04-107, Issue 2. European Space Agency. April.

ESA. (1993). ***"Telecommand decoder specification"***. ESA PSS-04-151, Issue 1. European Space Agency. September.

Ferriol P. et. al. (1998). ***"A double CAN architecture for fault-tolerant control systems"***. Proceedings of the 5th International CAN conference. San Jose. California. USA. November.

Ford S. (1999). ***"Satellite Roundup"***. Article published in *QST* (Official journal of the American Radio Relay League). Vol. 83, no.9. September.

Halsall F. (1992). ***"Data communications, Computer Networks and Open Systems"***. Addison-Wesley Publishers. USA.

IRIG. (1999). ***"Telemetry Standards"***. IRIG Standard 106-99. Telemetry Group (Range Commanders Council). U.S. Army White Sands Missile Range, New Mexico. January.

ISO 11898. (1993). ***"Road vehicles - Interchange of digital information - Controller Area Network (CAN) for high speed communication"***. International standard. First edition. November.

Jilla C.D. (1997). ***"Satellite design: Past, Present and Future"***. International journal of Small Satellite Engineering, Issue 1, February.

Johns A.T. and Krimchansky A. (1997). ***"EDOS experiences with CCSDS in support of the EOS spacecraft"***. Proceedings of the International Telemetering Conference.

Las Vegas. Nevada, USA. October.

Johnson B.W. (1989). ***"Design and Analysis of Fault Tolerant Digital Systems".*** Addison-Wesley Publishers. USA.

Jones G. (1995). ***"Introduction to packet radio".*** Published in 'Packet radio: What? Why? How? - Articles and information on general packet radio topics'. Tucson Amateur Packet Radio (TAPR) publication #95-1.

Kasser J. (1992). ***"Amateur satellite telemetry: past, present and future".*** Proceedings of the National Telesystems Conference, Reston, VA. USA.

Koekemoer J.A. (1997). ***"Reliable telemetry and telecommand in LEO satellites".*** Proceedings of the 6th Teletraffics Conference, Grahamstown, South Africa, September.

Koekemoer J.A and Bakkes P.J. (1999). ***"Distributed Telemetry & Telecommand for future SUNSAT missions".*** Acta Astronautica (Journal of the International Academy of Astronautics). Volume 44. No. 7-12. Page 531-541. September.

Larson W.J., and Wertz J.R. (1992) ***"Space Mission Analysis and Design".*** Microcosm Inc., Kluwer Academic Publishers, London.

Lawrenz W. (1997). ***"CAN System Engineering - from theory to practical applications".*** Springer-Verlag. New York.

Le Roux A.G. (1995). ***"Software for the SUNSAT Serial Bus and architectural description".*** ESL internal document. DCI 5.26/3/7 & DCI 5.26/3/2. University of Stellenbosch. October.

Le Roux A.G. (1998). ***"An FPGA-based approach to the compacting of the SUNSAT-1 TTM system".*** Project report presented in partial fulfillment of the requirements for the degree in Master in Engineering (Electronic) at the University of Stellenbosch.

Mansi L.S.A. and Sweeting M.N. (1987). ***"The UOSAT-2 spacecraft telemetry and telecommand subsystems".*** Journal of the institution of Electronic and Radio

Engineers, Sept/Oct.

Milne G.W., Koekemoer J.A. et. al. (1999) ***"SUNSAT - Launch and first six months orbital performance".*** Proceedings of the 13th annual AIAA/USU Conference on Small Satellites. Utah, USA, August.

Milne G.W. et al. (1999). ***"Early orbit status of SUNSAT, South Africa's first satellite ."*** Proceedings of Unispace 3, Vienna, Austria, July.

Milne G.W., Koekemoer J.A. et. al. (1998). ***"EMC and RFI problems and solutions on the SUNSAT micro satellite".*** South African Symposium on Antennas and Propagation/Microwave Theory and Techniques. Cape Town.

Mostert S. and Koekemoer J.A. (1997). ***"The science and engineering payloads and experiments on SUNSAT".*** Acta Astronautica (Journal of the International Academy of Astronautics). Volume 41. No. 4-10. Page 401-411. Aug.-Nov.

Pisacane V.L. and Moore R.C. (1994). ***"Fundamentals of Space Systems".*** Oxford University Press, Oxford.

Rabbie H. (1998). ***"Implementing the LonTalk protocol for Intelligent Distributed Control".*** Embedded Systems Conference. San Jose, California, USA. November.

Romkey J. (1988). ***"A nonstandard for transmission of IP datagrams over serial lines: SLIP".*** Network Working Group. Computer and Information Science. The Ohio State University. June.

Sinander P. and Habinc S. (1994). ***"Components for building telecommand and telemetry on-board systems".*** Proceedings of the On-board Data Management Symposium, Rome, January.

Steenkamp N.L. (1996). ***"The instrumentation bus protocol and the bus manager & bus driver SW structures".*** ESL internal document. DCI 5.6.1, DCI 5.7.3, DCI 5.7.4. University of Stellenbosch. May.

Szydlowski C. (1994). ***"Tradeoffs between stand-alone and integrated CAN***

peripherals". Proceedings of the First International CAN Conference. Mainz. Germany. September.

Tindell K. and Burns A. (1994). ***"Guaranteeing message latencies on Control Area Network (CAN)"***. Proceedings of the First International CAN conference. Mainz. Germany. September.

Tompkins S., et. al. (1997). ***"Lessons learned from using space data systems standards in flight missions"***. Proceedings of the International telemetering conference. Session XVIII. Las Vegas, Navada, USA. October.

Weber D.M. (1992). ***"SUNSAT Technical Specification. Section 3.5 & 5.27: Command Coding"***. Internal document, Electronic Systems Laboratory, University of Stellenbosch, August.

Wolf A and Koller C. (1998). ***"16-bit microcontroller with two on-chip CAN modules"***. Proceedings of the 5th International CAN conference. San Jose. California. USA. November.

Wong T. (1999). ***"Understanding USB"***. Article published in 'Electronics World'. Volume 105. Number 1763. November.

Index